BNAIC/BeneLearn 2021

33rd Benelux Conference on Artificial Intelligence and
30th Belgian-Dutch Conference on Machine Learning

AI in Action

FACULTY OF SCIENCE, TECHNOLOGY AND MEDICINE

LUXEMBOURG INSTITUTE OF SCIENCE AND TECHNOLOGY

Luxembourg National Research Fund

BNVKI AIABN

Proceedings of BNAIC/BeneLearn 2021

33rd Benelux Conference on Artificial Intelligence and
30th Belgian-Dutch Conference on Machine Learning

November 10–12, 2021
Belval, Esch-sur-Alzette (Luxembourg)
https://bnaic2021.uni.lu/

Editors
Luis A. Leiva, Cédric Pruski, Réka Markovich, Amro Najjar, Christoph Schommer

Organized by the University of Luxembourg, under the auspices of the Faculty of Science, Technology, and Medicine (FSTM) and the Interdisciplinary Lab for Intelligent and Adaptive Systems (ILIAS), and the IT for Innovative Services (ITIS) research department from the Luxembourg Institute of Science and Technology (LIST).

Sponsored by Luxembourg's National Research Fund (FNR), the Dutch Foundation for Neural Networks (SNN), the Foundation for Knowledge-Based Systems (SKBS), and the Benelux Association for AI (BNVKI).

ISSN
2799 2527

Preface

Welcome to the proceedings of BNAIC/BeneLearn'21, the 33rd edition of the annual Benelux Conference on Artificial Intelligence, and the 30th edition of the annual Belgian-Dutch Conference on Machine Learning.

In 2021, this joint conference has been organized by the University of Luxembourg, under the auspices of the Faculty of Science, Technology, and Medicine (FSTM) and the Interdisciplinary Lab for Intelligent and Adaptive Systems (ILIAS), and the IT for Innovative Services (ITIS) research department from the Luxembourg Institute of Science and Technology (LIST).

Held yearly, the objective of BNAIC/BeneLearn is to promote and disseminate recent research developments in Artificial Intelligence in the Benelux. In 2021 we have come back to in-person attendance, under CovidCheck regulations, as a three-day event: from Wednesday 10 to Friday 12 November 2021.

BNAIC/BeneLearn 2021 has included invited keynote speakers, research presentations, posters, and demonstrations. The conference has provided ample opportunity for synergies and interaction between academia and industry. This year, the chosen motto of the conference was "AI in ACTION", to reflect the aforementioned synergies and interactions between academia and industry.

For the scientific part, we have welcomed four types of contributions, namely: A) regular papers, B) encore abstracts, of already published work in 2021, C) poster and demonstrations, and D) thesis abstracts. We received 105 submissions overall, out of which 98 were selected for presentation at the conference: 39 regular papers, 28 encore abstracts, 10 poster and demonstrations, and 21 thesis abstracts. Then, 14 regular papers were selected for inclusion in a post-proceedings volume of the Springer CCIS series, after a second round of reviewing by members of the program committee. All regular papers, posters, and demonstrations received three expert single-blind reviews on average, whereas thesis and encore abstracts were reviewed by at least one program committee member.

All scientific contributions were presented as 20-minute talks, for which the conference program comprised 4 parallel tracks. In addition to these sientific presentations, we had keynote presentations by Fosca Giannotti (ISTI-CNR Pisa, Italy), Katie Atkinson (University of Liverpool, UK), Carles Sierra (IIIA of CSIC, Spain), Manuela Naveau (Kunstuniversität Linz, Austria), Julie Bernauer (NVIDIA Corporation, USA), and Iris von der Tuin (Utrecht University). We also held a special FACt (FACulty focusing on the FACts of AI) session with presentations by

Benoit Macq (Polytechnic School of UCLouvain, Belgium), Gilles Louppe (University of Liège, Belgium), and Christoph Schommer (University of Luxembourg, Luxembourg).

To conclude, we want to express our gratitude to everyone who made this conference possible. Without their efforts, this conference could not have taken place. In addition to all invited speakers mentioned above, many thanks to our sponsors: Luxembourg's National Research Fund (FNR), the Dutch Foundation for Neural Networks (SNN), the Foundation for Knowledge-Based Systems (SKBS), and the Benelux Association for AI (BNVKI). We also thank all the organizing and program committee members for their hard work to guarantee the high quality of this conference, both before and during the conference. We also wish to thank all student volunteers, administrative, and secretarial assistants, and of course all the academic as well as business sponsors. Finally, we also thank all the authors who made important contributions to the conference.

Luis A. Leiva,
Cédric Pruski,
Réka Markovich,
Amro Najjar,
Christoph Schommer

Organization

General chairs

Thibaud Latour — LIST, University of Luxembourg

Leon van der Torre — ILIAS, University of Luxembourg

Program chairs

Luis A. Leiva — ILIAS, University of Luxembourg
General track

Cédric Pruski — LIST, University of Luxembourg
AI & Systems track

Reka Markovich — ILIAS, University of Luxembourg
AI & Law track

Amro Najjar — ILIAS, University of Luxembourg
AI & Ethics track

Christoph Schommer — ILIAS, University of Luxembourg
AI & Art Track

Local chairs

Cédric Pruski — LIST, University of Luxembourg

Amro Najjar — ILIAS, University of Luxembourg

Sviatlana Höhn — SaToSS, University of Luxembourg

Publicity chair

Vladimir Despotovic — ILIAS, University of Luxembourg

Website manager

Nina Hosseini-Kivanani — ILIAS, University of Luxembourg

Program Committee

V. Javier Traver — Universitat Jaume I, Spain

Frank vanHarmelen — Vrije Universiteit Amsterdam, Netherlands

Jefrey Lijffijt — Ghent University, Belgium

Michel Klein — Vrije Universiteit Amsterdam, Netherlands

Sven Mayer — LMU Munich, Germany

Verónica Romero — Universitat de València, Spain

Marija Slavkovik — University of Bergen, Norway

Fatiha Saïs — LRI, Université Paris Sud, France

Jennifer Spenader — University of Groningen, Netherlands

Lu Cao — Leiden University, Netherlands

Mateusz Dubiel — University of Luxembourg

Hui Wang — Leiden University, Netherlands

Yingqian Zhang — Eindhoven University of Technology, Netherlands

Marieke van Vugt — University of Groningen, Netherlands

Chiara Boldrini — CNR, Italy

Jana Koehler — DFKI, Germany

Stephan Sigg — Aalto University, Finland

Mitra Baratchi — Leiden University, Netherlands

Yolanda Spinola — University of Seville, Spain

Siham Tabik — University of Granada, Spain

Bart Bogaerts — Vrije Universiteit Brussel, Belgium

Egon L. van den Broek — Utrecht University, Netherlands

Peter Lucas — Leiden University, Netherlands

Nanne van Noord — University of Amsterdam, Netherlands

Walter Kosters — Leiden University, Netherlands

Henry Prakken — Utrecht University, Netherlands

Gerasimos Spanakis — Maastricht University, Netherlands

Frans Oliehoek — TU Delft, Netherlands

Tom Lenaerts — Vrije Universiteit Brussel, Belgium

Nicolas Gillis — Université de Mons, Belgium

Jan Lemeire — Vrije Universiteit Brussel, Belgium

Gilles Louppe — University of Liège, Belgium

Johan Kwisthout — Radboud Universiteit, Netherlands

Bert Bredeweg — University of Amsterdam, Netherlands

Aske Plaat — Leiden University, Netherlands

Tibor Bosse — Radboud Universiteit, Netherlands

Jef Wijsen — University of Mons, Belgium

John-Jules Meyer — Utrecht University, Netherlands

Floris Bex — Utrecht University, Netherlands

Mark Hoogendoorn — Vrije Universiteit Amsterdam

John A. Lee — UC Louvain, Belgium

Joost Vennekens — KU Leuven, Belgium

Arnoud Visser — Universiteit van Amsterdam, Netherlands

Yvan Saeys — Ghent University, Belgium

Mark H. M. Winands — Maastricht University

Remco Veltkamp — Utrecht University, Netherlands

Hendrik Blockeel — KU Leuven, Belgium

Peter van der Putten — Leiden University, Netherlands

Emma Frid — IRCAM, France

Dirk Thierens — Utrecht University, Netherlands

Wannes Meert — KU Leuven, Belgium

Tom Claassen — Radboud Universiteit, Netherlands

Walter Daelemans — University of Antwerp, Belgium

Mehdi Dastani — Utrecht University, Netherlands

Jonas Soenen — KU Leuven, Belgium

Ad Feelders — Utrecht University, Netherlands

Tim van Erven — University of Amsterdam, Netherlands

Arjen Hommersom — Open University of the Netherlands, Netherlands

Menno van Zaanen — North-West University, South Africa

Vlado Menkovski — Eindhoven University of Technology, Netherlands

Sebastijan Dumancic — TU Delft, Netherlands

Tom Heskes — Radboud Universiteit, Netherlands

Miguel A. Ferrer — ULPGC, Spain

Contents

Regular papers

BNAIC/BeneLearn proceedings
November 10–12, 2021
Belval, Esch-sur-Alzette (Luxembourg)

The Effect of Noise Level on the Accuracy of Causal Discovery Methods with Additive Noise Models

Benjamin Kap[0000−0002−9230−9341], Marharyta Aleksandrova[0000−0002−1863−0129], and Thomas Engel[0000−0002−7374−3927]

University of Luxembourg
2, avenue de l'Université
L-4365 Esch-sur-Alzette
{benjamin.kap, marharyta.aleksandrova, thomas.engel}@uni.lu

Abstract. In recent years a lot of research was conducted within the area of causal inference and causal learning. Many methods were developed to identify the cause-effect pairs. These methods also proved their ability to successfully determine the direction of causal relationships from observational real-world data. Yet in bivariate situations, causal discovery problems remain challenging. A class of methods, that also allows tackling the bivariate case, is based on Additive Noise Models (ANMs). Unfortunately, one aspect of these methods has not received much attention until now: *what is the impact of different noise levels on the ability of these methods to identify the direction of the causal relationship?* This work aims to bridge this gap with the help of an empirical study. We consider a bivariate case and two specific methods *Regression with Subsequent Independence Test* and *Identification using Conditional Variances*. We perform a set of experiments with an exhaustive range of ANMs where the additive noises' levels gradually change from 1% to 10000% of the causes' noise level (the latter remains fixed). Additionally, we consider several different types of distributions as well as linear and non-linear ANMs. The results of the experiments show that these causal discovery methods can fail to capture the true causal direction for some levels of noise.

Keywords: Causal Learning · Additive Noise Models · Noise Level.

1 Introduction

Thanks to the technological and computational advances during the last decades, scientists were able to tackle successfully non-trivial problems from different research areas, with causality being a prominent example. One of the fundamental problems of causality theory is to determine the causal relationship between two or more variables. This problem is known as *causal discovery*, *causal identification* or *structure learning* [8, 27]. For example, given altitude and temperature, we want to answer the question if the temperature has an effect on altitude, or if

altitude has an effect on temperature. This is of particular interest since if such a causal relationship is known then one can predict the effects on a system in case of an intervention or a perturbation.

Controlled experimentation, or A/B tests, are considered to be a golden standard for causal discovery [11, 34]. In such experiments, there are two identical groups with only one variation. The only variable that is varied (intervened on) is the potential cause. This procedure allows estimating the causal effect of this variable in a given system. A/B tests are widely used in practical applications. For example, testing the efficacy of medications is usually done with A/B tests, see [32] for an example. In this case, the first group, also known as *control group*, receives no medication or a placebo, and the second group, known as *intervention group*, receives the real medication. The results show the true effect (if any) of the medication on human health. However, such tests are often too expensive, unethical, or even technically impossible to execute. For example, to test the effect of smoking on health with this approach, one needs two non-smoker groups. Next, the members of one group should be forced to smoke, and the others not do so. Therefore, it is of great interest to determine causal relationships from observational data only.

There exist many methods which are able to determine causal relationships from observational data. One particular group of such methods is based on *Additive Noise Models* (ANMs). These methods, as the name suggests, exploit the additivity of the random hidden noise. ANMs received a lot of attention as they are well established and yielded many good results [12]. Despite all the research in the past years, one small but nonetheless important aspect of causal discovery with ANMs has not received much attention: how do different noise *levels* of the additive noise impact the correctness of these methods? In the real world, it can occur that noise levels change drastically from cause to effect. It can happen, for example, if the data collection process has a lot of interference like in outer space.

In this work, we aim to bridge this research gap with an empirical study. For our analysis, we selected two specific methods: *Regression with Subsequent Independence Test (Resit)* [20] and *Identification using Conditional Variances (Uncertainty Scoring)* [17]. We chose Resit, as it is known to produce reliable results [15]. However, this method is not capable to identify the correct causal direction in the case both the cause and the noise are Gaussian. In fact, this case was only recently successfully tackled by the Uncertainty Scoring method. That is why we chose the latter one as well. We perform a set of experiments with an exhaustive range of ANMs where the additive noises' levels gradually change from 1% to 10000% of the causes' noise level (the latter remains fixed). We also consider several types of distributions as well as linear and non-linear data. The results of the experiments show that these causal discovery methods can fail to capture the true causal direction for some levels of noise.

This paper is organized as follows. In Section 2 we introduce related work. Next, in Section 3 we describe the chosen causal discovery methods. In Section 4 and Section 5 we discuss the experimental setup and the experimental results

respectively. Lastly, in Section 6 we draw conclusions and present possible future work.

2 Related Work

Structure learning is the procedure of determining causal relationship directions from observational data only and representing these as a (causal) graph. The basic idea emerged from [33] as *path analysis*.

Judea Pearl presented in his work [8] a comprehensive theory of causality and unified the probabilistic, manipulative, counterfactual, and structural approaches to causation. From this work we have the following key point. If there is a statistical association, e.g. two variables X and Y are dependent, then one of the following is true: 1) there is a causal relationship, either X has an effect on Y or Y has an effect on X; 2) there is a common cause (*confounder*) that has an effect on both X and Y; 3) there is a possibly unobserved common effect of X and Y that is conditioned upon data acquisition (selection bias); or 4) there can be a combination of these. From there on, a lot of research has been conducted to develop theoretical approaches and methods for structure learning. In the rest of this section, we first introduce the common concept behind all these approaches, and then we present some major works related to additive noise models.

In general, all methods for structure learning exploit the complexity of the marginal and conditional probability distributions in some way, see [1–7, 9, 13, 14, 16, 18–25, 27–30, 35]. Under certain assumptions, these methods are then able to solve the task of causal discovery. Let C denote the cause and E the effect. Then their joint density can be expressed with $p_{C,E}(c, e)$. This joint density can be factorized into either (1) $p_C(c) \cdot P_{E|C}(e|c)$ or (2) $p_E(e) \cdot P_{C|E}(c|e)$. The idea is then that (1) gives models of lower total complexity than (2) and this allows us to conclude the causal relationship direction. Intuitively, this makes sense, because the effect contains information from the cause but not vice-versa (of course, under the assumption that there are no cycles aka feedback loops). Therefore, (2) has at least as much complexity as (1). However, the definition of complexity is ambiguous. For example, one can say that "p_C *contains no information about* $P_{E|C}(e|c)$" and then draw partial conclusions about the causal direction in a given system. This complexity question is often colloquially referred to as *breaking the symmetry*, that is $p_C(c) \cdot P_{E|C}(e|c) \neq p_E(e) \cdot P_{C|E}(c|e)$.

As it was already mentioned, causal discovery based on ANMs was widely studied in the research literature. Silva et al. introduced in [26] a method for learning the structure of linear latent variable models. The main assumption in their work is that each variable is a linear function of its parents plus an additive error term of positive finite variance. Hoyer et al. generalized the linear framework of additive noise models to the nonlinear case [4]. Earlier works often assumed linear models for continuous variables. The authors showed that if data contains non-Gaussian variables, then this can help in distinguishing the causal directions and identifying the causal graph. Mooij et al. introduced Resit[1]

[1] Resit method is described in Section 3.2.

method in [13]. This method is based on the idea of minimizing the statistical dependence between the regressors and residuals[2]. The authors demonstrated that if the residuals are no longer dependent on the input, then regression can successfully model the causal dependence. This method does not need to assume a particular distribution of the noise because any form of regression can be used (e.g., Linear Regression), and it is well suited for the task of causal inference in additive noise models. Next, Mooij et al. introduced a method to determine the causal relationship in cyclic additive noise models and showed that such models are generally identifiable in the bivariate, Gaussian-noise case [14]. Their method works for continuous data and can be seen as a special case of nonlinear independent component analysis. Later, Peters and Bühlmann proved in [19] *full identifiability*[3] of linear Gaussian structural equation models if all the noise variables have the same variance. In the next work, Peters et at. proposed a method that can identify the directed acyclic graph from the distribution under mild conditions [20]. In contrast, previous methods assumed faithfulness and could only identify the Markov equivalence class of the graph[4]. Finally, the authors of [1, 18] proved that linear Gaussian models with different error variance can be also identifiable. In their method, referred to as Uncertainty Scoring[5], this is done by ordering variables according to the law of total variances and then performing independence tests between them. Park extended this result to additive noise models in [17].

As we can see, many researchers contributed to the development of ANMs-based causal discovery methods and widened our understanding of their application cases. However, no previous research work analyzed how the level of noise variance relative to that of the cause variance can impact the accuracy of these methods. This question forms the basis of the current study.

3 Causal Discovery Methods

In this section, we introduce notations and then describe to two analyzed causal discovery methods: *Regression with Subsequent Independence Test (Resit)* [20], see Section 3.2, and *Identification using Conditional Variances (Uncertainty Scoring)* [17], see Section 3.3.

3.1 Notations

In the following text, we give a short definition of additive noise models for the bivariate case. For more details and multivariate cases, please refer to [4, 20].

[2] The residuals are defined as the difference between the actual output and the predicted output.

[3] *Full identifiability* means that not only the skeleton of the causal graph is recoverable but also the arrows are.

[4] *Markov equivalence class* refers to the class of graphs in which all graphs have the same skeleton.

[5] Uncertainty Scoring method is described in Section 3.3.

Let $X, Y \in \mathbb{R}$ be the cause and the effect respectively. Let there also be m latent (hidden) causes $U = (U_1, \ldots, U_m) \in \mathbb{R}^m$. Then the causal relationship can be modeled as follows.

$$\begin{cases} Y = f(X, U_1, \cdots, U_m) \\ X \perp\!\!\!\perp U \end{cases}, \text{ with } X \sim p_X(x) \text{ and } U \sim p_U(u_1, \cdots, u_m),$$

where $f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}$ is a linear or nonlinear function, and $p_X(x)$ and $p_U(u_1, \cdots, u_m)$ are the joint densities of the observed cause X and the latent causes U. We assume that there is no confounding, no selection bias, and no feedback loops between X and Y. In this case, X and U are independent, which is denoted by $X \perp\!\!\!\perp U$. Since the latent causes U are unobserved, their influence can be summarized with a single noise variable $N_y \in \mathbb{R}$, and the model can be rewritten as follows:

$$\begin{cases} Y = f(X, N_y) \\ X \perp\!\!\!\perp N_y \end{cases}, \text{ with } X \sim p_X(x) \text{ and } N_y \sim p_{N_y}(n_y).$$

In our experiments, we are considering both linear and nonlinear additive noise models:

$$Y = \beta X + N_y \text{ with } \beta \in \mathbb{R}, \text{ for the linear case}$$

and

$$Y = \beta X^\alpha + N_y \text{ with } \beta, \alpha \in \mathbb{R}, \text{ for the nonlinear case.}$$

Also, X and N_y can be drawn from one of the following three distributions: the normal distribution denoted by the calligraphic letter $\mathcal{N}$, the uniform distribution denoted by the calligraphic letter $\mathcal{U}$, or the Laplace distribution denoted by the calligraphic letter $\mathcal{L}$. For example, throughout this work "*X is drawn from a normal distribution*" is denoted by $X \sim \mathcal{N}$ or $X \sim \mathcal{N}(\mu_x, \sigma_x)$ with μ_x standing for the mean and σ_x for the standard deviation.

3.2 Regression with Subsequent Independence Test (Resit)

We implement Resit following Algorithm 1 from [15]. This algorithm requires the following inputs: X and Y, a regression method, and a score estimator $\hat{C} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$; it outputs *dir* (casual relationship **dir**ection). The idea is to regress Y on X, predict $\hat{Y}$, and then calculate residuals $Y_{res} = \hat{Y} - Y$. Y_{res} and X are then used to calculate $\hat{C}_{X \to Y}$, a score for the assumed case $X \to Y$. Similarly, to test the other causal direction ($Y \to X$), we regress X on Y, calculate residuals $X_{res} = \hat{X} - X$ and estimate $\hat{C}_{Y \to X}$. In our experiments, the generated data always follows $X \to Y$. This verifies the **assumption** that only one direction in our data is correct (and not both). Under this assumption, we can compare both scores directly to decide on the cause-effect direction, and we do not need to determine the value of α for the independence tests, see Eq. (1). Additionally, we can also use entropy estimators to estimate the score $\hat{C}$.

Algorithm 1 General procedure to decide whether $p(x, y)$ satisfies Additive Noise Model $X \to Y$ or $Y \to X$.

Input:
- I.i.d. sample data X and Y
- Regression method
- Score estimator $\hat{C} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$

Output:
- *dir*

$reg_1 \leftarrow$ Regress Y on X
$reg_2 \leftarrow$ Regress X on Y

$Y_{res} \leftarrow reg_1.predict(X) - Y$
$X_{res} \leftarrow reg_2.predict(Y) - X$

$\hat{C}_{X\to Y} \leftarrow \hat{C}(X, Y_{res})$
$\hat{C}_{Y\to X} \leftarrow \hat{C}(Y, X_{res})$

$$\textbf{return } dir = \begin{cases} X \to Y & \text{if } \hat{C}_{X\to Y} < \hat{C}_{Y\to X}, \\ Y \to X & \text{if } \hat{C}_{X\to Y} > \hat{C}_{Y\to X}, \\ ? & \text{if } \hat{C}_{X\to Y} = \hat{C}_{Y\to X}. \end{cases} \tag{1}$$

In Algorithm 1, it is possible to split the data into training and test parts. In this case, the training data is used to fit the regression model and the test data is used to calculate the value of $\hat{C}$. This procedure is referred to as *decoupled estimation* [12]. The advantage of splitting the data lies in the reduction of the computational time for calculating independence estimates $\hat{C}$. However, in this work, we use *coupled estimation*. This means that the entire data-set is used for both the regression and the independence estimation steps. The latter approach tends to produce more accurate results for independence estimation.

In our work, we use Linear Regression as a regression algorithm. If an appropriate transformation of coordinates is applied, Linear regression can be used in the non-linear cases as well. In our experiments, we used six different independence tests and six different entropy measures for calculating $\hat{C}$. In general, for the independence tests we have:

$$\hat{C}(X_{Test}, Y_{res}) = I(X_{Test}, Y_{res}),$$

with $I(\cdot, \cdot)$ being any independence test. In the case of entropy estimators we have:

$$\hat{C}(X_{Test}, Y_{res}) = H(X_{Test}) + H(Y_{res}),$$

with $H(\cdot)$ being any entropy measure. The entropy-based estimator score is derived from Lemma 1 in [12].

The following estimators were used in this work. The implementation of estimators with numbers 2 - 12 was taken from the *information theoretical estimators*

toolbox [31]. Here we briefly introduce every estimator. Mathematical formulas for each of them can be found in the Appendix.

1. *HSIC*: Hilbert-Schmidt Independence Criterion with RBF Kernel [6].
2. *HSIC_IC*: Hilbert-Schmidt Independence Criterion using incomplete Cholesky decomposition[7].
3. *HSIC_IC2*: Same as HSIC_IC but with lower precision.
4. *DISTCOV*: Distance covariance estimator using pairwise distances.
5. *DISTCORR*: Distance correlation estimator using pairwise distances. It is simply the standardized version of the distance covariance.
6. *HOEFFDING*: Hoeffding's Phi.
7. *SH_KNN*: Shannon differential entropy estimator using kNNs (k-nearest neighbors) where $k = 3$.
8. *SH_KNN_2*: Same as SH_KNN but with different search method.
9. *SH_KNN_3*: Same as SH_KNN but with $k = 5$.
10. *SH_MAXENT1*: Maximum entropy distribution-based Shannon entropy estimator.
11. *SH_MAXENT2*: Same as SH_MAXENT1 with minor changes.
12. *SH_SPACING_V*: Shannon entropy estimator using Vasicek's spacing method.

3.3 Identification using Conditional Variances (Uncertainty Scoring)

The Uncertainty Scoring method is composed of Algorithm 2 and Algorithm 3 from [17]. It consists of two parts: 1) ordering and 2) conditional independence testing.

For the first step, ordering, we used *backward step-wise selection* (Algorithm 2), as it is more convenient for implementation. The algorithm starts with a set S which contains all variables represented as nodes in a causal graph. Next, we iterate over S, and for each node, we calculate its conditional variance given all other remaining nodes. Then, we select the node with the highest conditional variance, append it to the ordering π, and also remove it from the set S. With the updated set S, we repeat this process until S is empty. Lastly, the *reverse* of the ordering π is returned. The first node to be appended to the ordering is the last one in the ordering, which is reflected in the name "*backward step-wise selection*".

In the second step, we perform uncertainty scoring using Algorithm 3. This algorithm iterates over the ordering π. For every node j, it performs conditional independence tests conditioning on every other node l appearing before the node j in the ordering π. If a node l is dependent on j, then it is added to the set of parents of j, denoted as $Pa(j)$. In this algorithm, the first node in the ordering never has parents, so the procedure starts with the second node. *Fisher's z-transform of the partial correlation,* is used for the conditional independence testing.

[6] Source: https://github.com/amber0309/HSIC

[7] Low rank decomposition of Gram matrices, which permits an accurate approximation to HSIC as long as the kernel has a fast decaying spectrum.

Algorithm 2 Backward step-wise selection

Input: All variables from an ANM: $X = (x_1, x_2, \ldots, x_n)$
Output: Estimated ordering $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$

Set $S = \{1, 2, \ldots, n\}$
List $\pi = [\ ]$
for $m = 1 \ldots n$ **do**
    for $j \in S$ **do**
        Estimate the conditional variance x_j given $\{x_1, \ldots, x_n\} \backslash x_j$, $\sigma^2_{j|S\backslash \mathrm{j}}$
    end
    Append $\pi_m = argmax_j \sigma^2_{j|S\backslash j}$ to π
    Update $S = S \backslash \pi_m$
end
return Reversed list π

Algorithm 3 Uncertainty Scoring

Input: All variables from an ANM: $X = (x_1, x_2, \ldots, x_n)$
Output: Dictionary with estimated parents for all variables: $G = \{Pa(x_1) : [\ldots], Pa(x_2) : [\ldots], \ldots, Pa(x_n) : [\ldots]\}$

Get ordering from backward step-wise selection: $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$
$G = \{\}$
for $m = 2 \ldots n$ **do**
    $Pa(\pi_m) = [\ ]$
    for $j = 1 \ldots m - 1$ **do**
        Conditional independence test between π_m and π_j given $\{\pi_1, \ldots, \pi_{m-1}\} \backslash \pi_j$
        If dependent, include π_j into $Pa(\pi_m)$
    end
    Insert $Pa(\pi_m)$ into G
end
return G

4 Experimental setup

Generation of synthetic data. For all empirical tests, we assume X to be a cause of Y, that is $X \to Y$. In the sense of additive noise models, we use the following equations: $Y = X + N_y$ for the linear case, and $Y = X^3 + N_y$ for the non-linear case, where

$$X \sim \begin{cases} \mathcal{N}(0,1) & \text{or} \\ \mathcal{U}(-1,1) & \text{or} \\ \mathcal{L}(0,1) \end{cases} \qquad \textit{and} \qquad N_y \sim \begin{cases} \mathcal{N}(0, 1 \cdot i) & \text{or} \\ \mathcal{U}(-1 \cdot i, 1 \cdot i) & \text{or} \\ \mathcal{L}(0, 1 \cdot i) \end{cases}$$

with i being a scaling factor for the noise level in N_y. The goal is to analyze how different standard deviations (boundaries for the uniform case) in the noise

term N_y relative to the standard deviations (or boundaries for the uniform case) in the X term impact the ANM methods.

To cover various dependencies between the distributions of X and N_y, we generate 199 different i factors:

$$i \in \{0.01, 0.02, \ldots, 1.00\} \cup \{1, 2, \ldots, 100\}.$$

For each i, every linear and non-linear combination with different distributions is tested. Totally, we have 18 combinations corresponding to the general structures $Y = X + N_y$ and $Y = X^3 + N_y$, where X and N_y are drawn from the three different distributions, $\mathcal{N}$, $\mathcal{U}$ or $\mathcal{L}$.

$$Y = X \sim \mathcal{N} + N_y \sim \mathcal{N},$$
$$Y = X \sim \mathcal{N} + N_y \sim \mathcal{U},$$
$$Y = X \sim \mathcal{N} + N_y \sim \mathcal{L},$$
$$\vdots$$
$$Y = X \sim \mathcal{L}^3 + N_y \sim \mathcal{L}.$$

Note that $\mathcal{L}^3$ here signifies the non-linear case $Y = X^3 + N_y$.

Evaluation. For each of the 18 combinations, we perform 100 tests. In every test, we generate 1000 new samples for X and N_y and attempt to identify the direction of the causal relationship[8] using one of the two algorithms presented in Section 3. Lastly, we simply calculate the fraction of successful tests and define this ratio as our accuracy measure.

5 Experimental Results

Since we used a large range for the values of i-factor, several different combinations of distributions, linear and non-linear data, we have too many results to show them all in detail in this paper. Therefore, we discuss several representative cases and provide a summary of all results. The latter shows for which values of i-factor the models are consistently identifiable. For the detailed analysis, we refer to the document [10]. Alternatively, all the results and source codes can be accessed from the relative repository[9].

5.1 Resit

We start with the analysis of Resit method. In this set of experiments, we are interested in which ranges of i-factor allow causal identifiability and how it is related to the functional model and the chosen independence estimator. Fig. 1

[8] The true direction of the causal relationship is known as we generate synthetic data.
[9] https://gitlab.com/Shinkaiika/noise-level-causal-identification-additive-noise-models

shows the detailed results for the following 4 linear combinations and their non-linear counterparts: $Y = \mathcal{N} + \mathcal{U}$, $Y = \mathcal{U} + \mathcal{N}$, $Y = \mathcal{U} + \mathcal{L}$, and $Y = \mathcal{L} + \mathcal{L}$. The y-axis shows the accuracy of causal discovery ($\frac{\#\text{successful tests}}{100}$), and the x-axis corresponds to i-factor. Different colors encode 12 estimators used in this work. The value of accuracy close to 0.5 means that Resit outputs the correct causal direction in only 50% of the tests thus indicating **unidentifiability.** The values close to 1 signify very good/consistent **identifiability**. In the following text, we analyze the results for individual models.

Fig. 1a shows the linear model $Y = \mathcal{N} + \mathcal{U}$. We can see, that all estimators reach an accuracy close to 100% inside the interval $i \in [0.8; 5]$. However, for smaller or larger i-factors the accuracy of all estimators start to drop until they reach unidentifiability (~ 0.5). Not all estimators perform the same. For example, HISC with Incomplete Cholesky decomposition performs worse for decreasing i-factors compared to all other estimators. SH_SPACING_V performs the best among all estimators for this linear model. Fig. 1b shows the non-linear model $Y = \mathcal{N}^3 + \mathcal{U}$. The non-linear version shows much better results. With $i \in [0.2; 100]$, we have accuracy close to 100% for all estimators. Only a few estimators drop towards unidentifiability for $i < 0.2$.

Fig. 1c shows the linear model $Y = \mathcal{U} + \mathcal{N}$. For $i \in [0.1; 1]$ this model is identifiable. However, for larger values of i-factor, the accuracy of many estimators drop quickly. In this range, SH_SPACING_V remains above 90%, most other estimators drop between 60% and 80% but HSIC_IC and HSIC_IC2 drop to 50% accuracy demonstrating complete unidentifiability. Fig. 1d shows the results for the non-linear version of this model. For $i \leq 1$, all estimators remain above 90% accuracy, with the exceptions now being HSIC_IC and HSIC_IC2. For i-factors larger than 1, estimators behave differently. SH_KNN, SH_KNN_2, SH_KNN_3, DISTCOV, DISCORR and HOEFFDING remain above 90% accuracy up to $i = 100$. SH_MAXENT1 remains between 80% and 90%, HSIC and SH_MAXENT2 between 60% and 80%, and HSIC_IC and HSIC_IC2 become unidentifiable.

Fig. 1e shows the linear case $Y = \mathcal{U} + \mathcal{L}$ and Fig. 1f shows the non-linear case $Y = \mathcal{U}^3 + \mathcal{L}$. The demonstrated results are quite similar to the two cases discussed above. This indicates that models with the same type of distribution for X behave similarly.

Fig. 1g shows the linear case $Y = \mathcal{L} + \mathcal{L}$. For $i \in [0.1; 10]$ most estimators are above 90%, except SH_KNN, SH_KNN_2 and SH_KNN_3 which are above 90% for $i \in [0.4; 2]$. For larger values of i-factor, all estimators drop quickly to unidentifiability. Finally, Fig. 1h shows the non-linear case $Y = \mathcal{L}^3 + \mathcal{L}$. Similarly to the model $Y = \mathcal{N}^3 + \mathcal{U}$ presented in Fig. 1b, this model demonstrates that non-linearity generally helps in identifying causal relationships. For $i \in [0.15; 100]$ all estimators are above 90% accuracy, often reaching 100%.

The experimental results for Resit with linear and non-linear models are summarized in Tables 1 and 2 respectively. The rows correspond to different estimators, and columns correspond to structural equation models. The values in the cells show on what range of i a particular estimator *can* reach over 90%

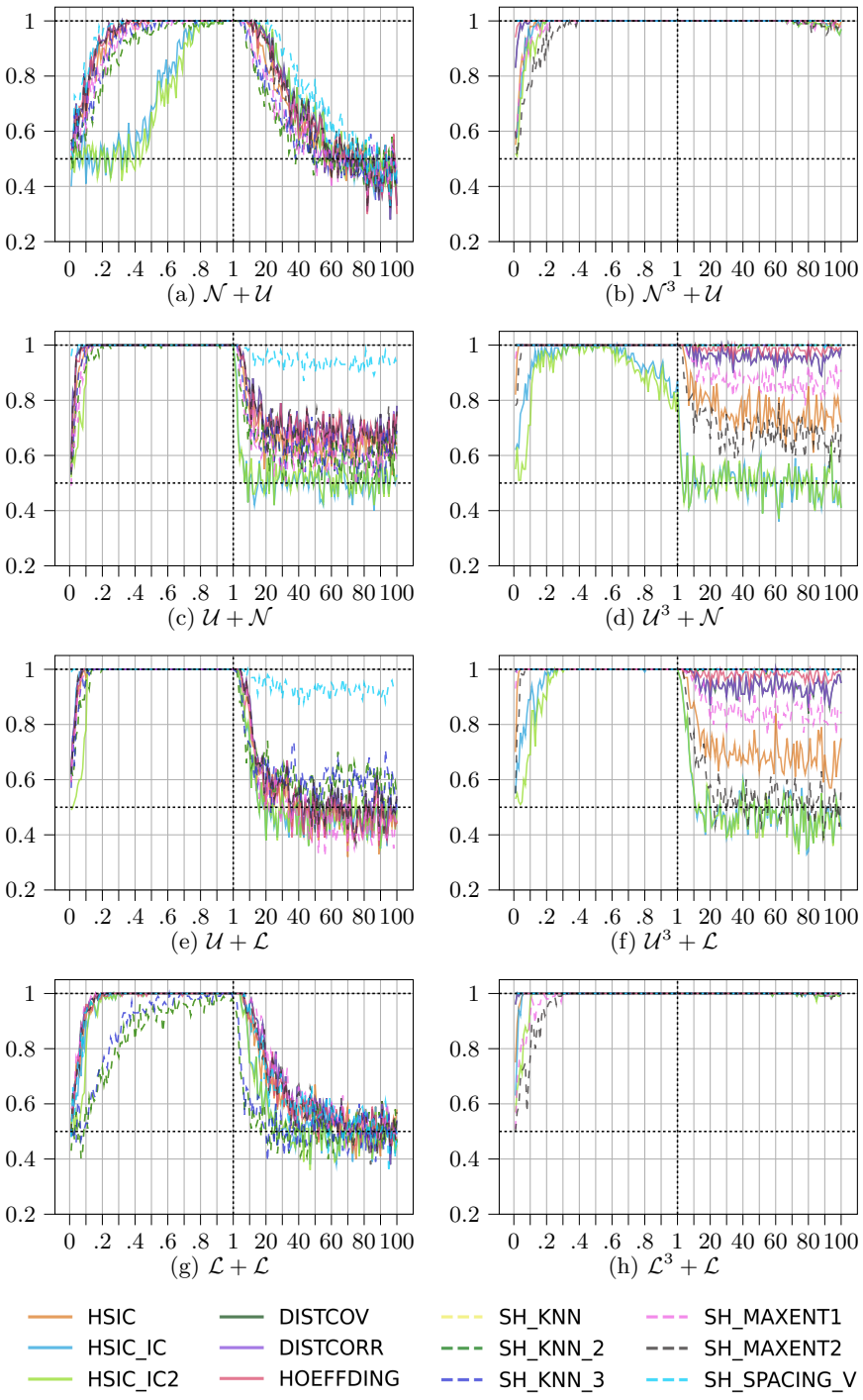


Fig. 1: Several selected detailed results for Resit. *x-axis shows the values of i-factor and y-axis shows the accuracy of causal identification.*

Table 1: Summary for Resit with linear models. The numbers reflect the ranges of i-factor that allow identifiability with accuracy around or above 90%.

Equation	$\mathcal{N}+\mathcal{N}$	$\mathcal{N}+\mathcal{U}$	$\mathcal{N}+\mathcal{L}$	$\mathcal{U}+\mathcal{N}$	$\mathcal{U}+\mathcal{U}$	$\mathcal{U}+\mathcal{L}$	$\mathcal{L}+\mathcal{N}$	$\mathcal{L}+\mathcal{U}$	$\mathcal{L}+\mathcal{L}$
HSIC		0.17 - 18	0.13 - 8	0.05 - 6	0.06 - 16	0.04 - 7	0.1 - 7	0.12 - 23	0.1 - 13
HSIC_IC		0.65 - 26	0.31 - 7	0.04 - 3	0.06 - 15	0.04 - 5	0.1 - 4	0.14 - 26	0.1 - 8
HSIC_IC2		0.7 - 26	0.33 - 7	0.1 - 3	0.14 - 15	0.11 - 5	0.1 - 4	0.14 - 26	0.12 - 8
DISTCOV		0.16 - 23	0.13 - 7	0.04 - 7	0.05 - 21	0.04 - 10	0.1 - 7	0.1 - 25	0.08 - 15
DISTCORR		0.16 - 23	0.13 - 7	0.04 - 7	0.05 - 21	0.04 - 10	0.1 - 7	0.1 - 25	0.08 - 15
HOEFFDING		0.16 - 25	0.13 - 8	0.04 - 7	0.05 - 21	0.04 - 8	0.1 - 7	0.1 - 25	0.1 - 10
SH_KNN		0.32 - 12	0.76 - 1	0.08 - 4	0.07 - 12	0.09 - 4	0.61 - 1	0.27 - 12	0.37 - 3
SH_KNN_2		0.32 - 12	0.76 - 1	0.08 - 4	0.07 - 12	0.09 - 4	0.61 - 1	0.27 - 12	0.37 - 3
SH_KNN_3		0.24 - 12	0.51 - 1	0.05 - 5	0.07 - 14	0.05 - 5	0.37 - 3	0.21 - 15	0.32 - 4
SH_MAXENT1		0.23 - 12	0.12 - 10	0.06 - 4	0.1 - 12	0.04 - 8	0.07 - 13	0.11 - 24	0.07 - 17
SH_MAXENT2		0.15 - 22	0.13 - 7	0.03 - 7	0.05 - 17	0.04 - 8	0.1 - 7	0.11 - 23	0.1 - 13
SH_SPACING_V		0.13 - 33	0.17 - 5	0.01 - 100	0.03 - 40	0.01 - 100	0.14 - 6	0.11 - 33	0.09 - 13

Table 2: Summary for Resit with non-linear data. The numbers reflect the ranges of i-factor that allow identifiability with accuracy around or above 90%.

Equation	$\mathcal{N}^3+\mathcal{N}$	$\mathcal{N}^3+\mathcal{U}$	$\mathcal{N}^3+\mathcal{L}$	$\mathcal{U}^3+\mathcal{N}$	$\mathcal{U}^3+\mathcal{U}$	$\mathcal{U}^3+\mathcal{L}$	$\mathcal{L}^3+\mathcal{N}$	$\mathcal{L}^3+\mathcal{U}$	$\mathcal{L}^3+\mathcal{L}$
HSIC	0.04 - 100	0.08 - 100	0.04 - 100	0.02 - 6	0.03 - 16	0.03 - 7	0.02 - 100	0.04 - 100	0.02 - 100
HSIC_IC	0.04 - 83	0.06 - 100	0.04 - 70	0.1 - 0.92	0.14 - 13	0.1 - 4	0.03 - 100	0.05 - 100	0.03 - 100
HSIC_IC2	0.08 - 83	0.08 - 100	0.09 - 70	0.12 - 0.91	0.17 - 13	0.17 - 4	0.7 - 100	0.07 - 100	0.09 - 100
DISTCOV	0.02 - 100	0.02 - 100	0.02 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100
DISTCORR	0.02 - 100	0.02 - 100	0.02 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100
HOEFFDING	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100
SH_KNN	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100
SH_KNN_2	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100
SH_KNN_3	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100
SH_MAXENT1	0.05 - 100	0.06 - 100	0.05 - 100	0.01 - 100	0.02 - 90	0.01 - 88	0.1 - 100	0.17 - 100	0.1 - 100
SH_MAXENT2	0.11 - 98	0.16 - 100	0.1 - 100	0.03 - 4	0.04 - 12	0.04 - 5	0.14 - 100	0.15 - 100	0.15 - 100
SH_SPACING_V	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100	0.01 - 100

accuracy. Estimators have some variance in the results and thus on some intervals they fall below 90% accuracy. The limits in the cells were chosen as follows: the lower limit designates where an estimator reaches 90% or higher for the first time, and the upper limit designates for which value of i it was observed for the last time. In between, most of the time estimators remain above 90% or rarely fall below, but never below 80% accuracy. An empty cell means that the corresponding estimator never resulted in accuracy $\geq 90\%$.

As the results show, different noise levels do have an impact on the identifiability performance of Resit. In general, the linear equation models are more fragile than the non-linear ones. This is explained by the fact that the non-linear relationships tend to break the symmetry between the variables easier, see [4]. The only structural equation which always remains unidentifiable is $Y = \mathcal{N}+\mathcal{N}$, see [24]. For all other cases, all estimators reach an accuracy of over 90% for some values of i-factor. For example, all estimators perform perfectly when the noise

level of the X term is comparable to the noise level of the corresponding noise term (N_y), that is $i = 1$. For other values of i, there are differences between linear and non-linear equations. Generally, the accuracy for linear cases drops if $i > 7$. However, most non-linear cases retain accuracy over 90% for much larger values of i-factor, even up to 100. Similar results are observed for the decreasing i-factors.

We can also observe differences between estimators in terms of accuracy. For example, HSIC is overall the best performing independence estimator while HSIC_IC and HSIC_IC_2 perform the worst. SH_SPACING_V is the best performing entropy estimator while SH_MAXENT1 and SH_MAXENT2 perform the worst. Some estimators show better performance for particular structural causal models, for example, SH_SPACING_V for $Y = \mathcal{U} + \mathcal{N}$; others are particularly unsuitable for some structural equations, for example, HSIC_IC and HSIC_IC2 for $Y = \mathcal{N} + \mathcal{U}$. For all non-linear equation models, SH_SPACING_V and the three Shannon kNN estimators result in accuracy close to 100% for all values of i. SH_SPACING_V also keeps its good performance in the case of linear equation models. As for independence measures, HSIC, DISTCOV, DISTCORR, and HOEFFDING perform quite similarly and are good overall. Note again, that these results are based on the assumption that in our bivariate structure only one direction of the causal relationship is present, namely $X \to Y$. Without this assumption, we cannot compare the estimates directly but rather need to compare the estimate to a derived p-value given some significance level α.

5.2 Uncertainty Scoring

Fig. 2 shows the results for the Uncertainty Scoring algorithm. Recall that for these experiments we use only one estimator, the Fisher's conditional independence test. Therefore, we use different colors and styles of lines to encode structural equation models. The colours of the lines correspond to the distribution type of the noise variable N_y with the following coding: blue for $N_y \sim \mathcal{N}$, green for $N_y \sim \mathcal{U}$, and red for $N_y \sim \mathcal{L}$. The type of the lines encodes the distribution type of the cause X as follows: solid line for $X \sim \mathcal{N}$, dashed line for $X \sim \mathcal{U}$, and dotted line for $X \sim \mathcal{L}$. As in the previous experiment, the x-axis shows the values of i-factor and the y-axis shows the accuracy of causal identification. However, the results should be interpreted differently. The Uncertainty Scoring method generates a set of parents for every variable. This set can be empty or can contain cause variables. Therefore, only one structure of this result is correct and thus the y-axis of the plots in Fig. 2 shows consistent identifiability at 1, and consistent unidentifiability at 0.

We proceed to the analysis of the results. First, we can notice that the linear Gaussian model $Y = \mathcal{N} + \mathcal{N}$ is now identifiable, as it was demonstrated by the authors of this method [17]. Interestingly, for this method, the linear cases perform better than the non-linear as opposed to Resit. Only the non-linear cases where the cause X is drawn from the Uniform distribution $\mathcal{U}$ show the same performance as the linear cases. This group of models demonstrates good identifiability for $i < 1$, however the accuracy drops fast for $i > 1$. The reason for

accuracy degradation lies within step 2 of the method, the conditional independence test. If noise levels are significantly different, then the independence test fails to capture the correlation between the two nodes and therefore concludes that the nodes are independent (Type II Error). However, for any given i, the ordering step always performs correctly[10].

We can also notice that models with similar structures have similar performance. For example, in Fig. 2b we can clearly identify 3 groups: 1) the group of dashed lines representing models with $X \sim \mathcal{U}$ show the best performance for $i < 1$ and the worst performance for $i > 1$; 2) the group of dotted lines corresponding to models with $X \sim \mathcal{L}$ demonstrate the worst accuracy for $i < 1$ and the best accuracy for $i > 1$; finally 3) the group of solid lines that represent the models with $X \sim \mathcal{N}$ lie in the middle. A similar observation was done for Resit as well, that is the type of the distribution of the cause variable affects the accuracy of causal discovery. If we analyze the linear cases from Fig. 2a in the same way, we can notice that here the type of the distribution of the noise variable N_y probably has more impact. Indeed, the lines overlap, but they are now grouped more by colors than by line type. Again, we can observe 3 groups: 1) the group of green lines corresponding to the models with $N_y \sim \mathcal{U}$ show worse performance for $i < 1$ and better performance for $i > 1$; 2) the group of red lines representing the models with $N_y \sim \mathcal{L}$ have better performance for $i < 1$ and worse accuracy for $i > 1$; 3) and the group of blue lines corresponding to $N_y \sim \mathcal{N}$ lies in between.

The results obtained for the Uncertainty Scoring method are summarized in Table 3. Here, each row corresponds to a combination of distribution types. The second and the third columns show the results for linear or non-linear models respectively. The values inside the table are encoded in the same way as it was done for Table 1; that is they show the ranges where the method has an accuracy around or above 90%.

6 Conclusions

The results from the experiments showed that two analyzed causal discovery methods, Resit and Uncertainty Scoring, are affected by different noise scales. For significantly small noise levels in the disturbance term N_y, or significantly high noise levels, these causal discovery methods fail to capture the true causal relationship of the given structural equation model. Recall that *significantly* here depends on the model. For example, for some models, if the noise level was already twice larger then the methods failed to determine the causal direction consistently. Other models remained identifiable with 100 times higher noise levels. The range of different noise levels analyzed in this work is quite exhaustive and realistically speaking having noise levels 100 times higher than the potential cause variable is very rare. Additionally, with very high noise levels the effect

[10] A quick test in python shell, with $i = 57$, $X \sim \mathcal{L}$ and $N_y \sim \mathcal{U}$ and 100 repetitions showed that in these runs the ordering was always correct but only in 35 runs (from the 100 repetitions) the independence tests were correct.

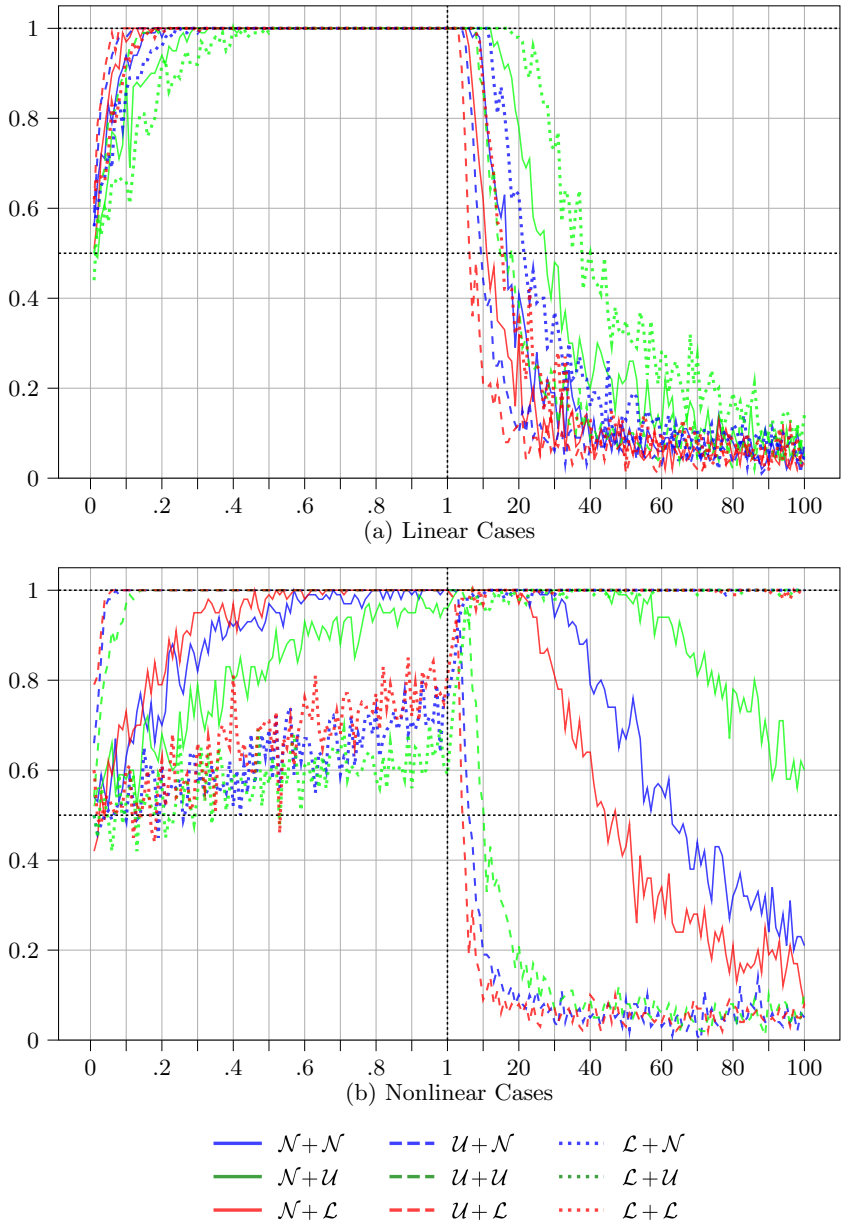


Fig. 2: Results of the Uncertainty Scoring algorithm. *x-axis shows the values of i-factor and y-axis shows the accuracy of causal identification.*

Table 3: Summary for Uncertainty Scoring. The numbers reflect the ranges of i-factor that allow identifiability with accuracy around or above 90%.

Equation	**Linear**	**Non-Linear**
$\mathcal{N}+\mathcal{N}$	0.08 - 10	0.33 - 37
$\mathcal{N}+\mathcal{U}$	0.16 - 10	0.52 - 67
$\mathcal{N}+\mathcal{L}$	0.05 - 6	0.23 - 25
$\mathcal{U}+\mathcal{N}$	0.04 - 5	0.04 - 4
$\mathcal{U}+\mathcal{U}$	0.1 - 8	0.05 - 6
$\mathcal{U}+\mathcal{L}$	0.03 - 3	0.03 - 3
$\mathcal{L}+\mathcal{N}$	0.14 - 13	4 - 100
$\mathcal{L}+\mathcal{U}$	0.19 - 26	5 - 100
$\mathcal{L}+\mathcal{L}$	0.1 - 10	2 - 100

of the cause variable is very likely negligible anyways. However, the discovered relationships can be useful to guide researchers in practical applications. We also observed different behavior for different distribution types (e.g., Gaussian or Uniform).

For both methods, we observed that if the variance of the noise term is smaller than that of the cause, then models remained identifiable. The opposite relationship is observed when the variance of the noise term is larger. For example, often when the standard deviation of the noise term was only half of that of the cause, the model was still identifiable. However, in several cases, if the standard deviation of the noise term was already twice larger than the standard deviation of the cause, then the model became unidentifiable. We also tested linear and non-linear models and our results show that non-linear models were still identifiable in situations where the linear models are not. For example, some non-linear models, where the noise term's variance was 100 times higher than that of the cause, were still perfectly identifiable while their linear counterparts were not.

Lastly, for Resit we used several estimators: 6 independence estimators and 6 entropy estimators. Our results show differences in terms of performance depending on which estimator is used. We observed that Hilbert-Schmidt Independence Criterion with RBF Kernel was the best independence estimator, and Shannon entropy estimator using Vasicek's spacing method was the best entropy estimator.

In our experiments, we tested only two particular methods and three different distribution types. However, similar results are expected for other methods of causal discovery with additive noise models, as their common failing point lies in the independence estimation.

Future work. In reality, observed data does not always strictly follow a certain distribution type. As there are many different possible combinations, it would be interesting to generalize the impact of different noise levels on any

distribution by using the different properties an observed distribution exhibits. Furthermore, this work does not formalize mathematically the effect of different noise levels in ANM causal discovery methods. This could be done in future work.

7 Acknowledgments

This work was partially supported by the European Union Horizon 2020 research programme within the project CITIES2030 "Co-creating resilient and sustainable food towards FOOD2030", grant 101000640.

Appendix

Detailed description of estimators

1. **HSIC**: Hilbert-Schmidt Independence Criterion with RBF Kernel [11]

$$I_{HSIC}(x,y) := ||C_{xy}||^2_{HS}$$

where C_{xy} is the cross-covariance operator and HS the squared Hilbert-Schmidt norm.
2. **HSIC_IC**: Hilbert-Schmidt Independence Criterion using incomplete Cholesky decomposition (low rank decomposition of the Gram matrices, which permits an accurate approximation to HSIC as long as the kernel has a fast decaying spectrum) which has $\eta = 1 * 10^{-6}$ precision in the incomplete cholesky decomposition.
3. **HSIC_IC2**: Same as HSIC_IC but with $\eta = 1 * 10^{-2}$.
4. **DISTCOV**: Distance covariance estimator using pairwise distances. This is simply the L^2_w norm of the characteristic functions φ_{12} and $\varphi_1\varphi_2$ of input x, y:

$$\varphi_{12}(\boldsymbol{u}^1, \boldsymbol{u}^2) = \mathbb{E}[e^{i\langle \boldsymbol{u}^1, \boldsymbol{x}\rangle + i\langle \boldsymbol{u}^2, \boldsymbol{y}\rangle}],$$
$$\varphi_1(\boldsymbol{u}^1) = \mathbb{E}[e^{i\langle \boldsymbol{u}^1, \boldsymbol{x}\rangle}],$$
$$\varphi_2(\boldsymbol{u}^2) = \mathbb{E}[e^{i\langle \boldsymbol{u}^2, \boldsymbol{y}\rangle}].$$

With $i = \sqrt{-1}$, $\langle \cdot, \cdot \rangle$ the standard Euclidean inner product and $\mathbb{E}$ the expectation. Finally, we have:

$$I_{dCov}(x,y) = ||\varphi_{12} - \varphi_1\varphi_2||_{L^2_w}$$

5. **DISTCORR**: Distance correlation estimator using pairwise distances. It is simply the standardized version of the distance covariance:

$$I_{dCor}(x,y) = \begin{cases} \frac{I_{dCov}(x,y)}{\sqrt{I_{dVar}(x,x) I_{dVar}(y,y)}}, & \text{if } I_{dVar}(x,x) I_{dVar}(y,y) > 0 \\ 0, & \text{otherwise,} \end{cases}$$

with

$$I_{dVar}(x,x) = ||\varphi_{11} - \varphi_1\varphi_1||_{L^2_w},\ I_{dVar}(y,y) = ||\varphi_{22} - \varphi_2\varphi_2||_{L^2_w}$$

(see characteristic functions under 4. DISTCOV)
6. **HOEFFDING**: Hoeffding's Phi

$$I_\Phi(x,y) = I_\Phi(C) = \left(h_2(d) \int_{[0,1]^d} [C(\boldsymbol{u}) - \Pi(\boldsymbol{u})]^2 d\boldsymbol{u} \right)^{\frac{1}{2}}$$

with C standing for the copula of the input and Π standing for the product copula.

[11] Source: https://github.com/amber0309/HSIC

7. **SH_KNN**: Shannon differential entropy estimator using kNNs (k-nearest neighbors)

$$H(\boldsymbol{Y}_{1:T}) = log(T-1) - \psi(k) + log(V_d) + \frac{d}{T}\sum_{t=1}^{T} log(\rho_k(t))$$

with T standing for the number of samples, $\rho_k(t)$ - the Euclidean distance of the k^{th} nearest neighbour of $\boldsymbol{y}_t$ in the sample $\boldsymbol{Y}_{1:T}\backslash\{\boldsymbol{y}_t\}$ and $V \subseteq \mathbb{R}^d$ a finite set.
8. **SH_KNN_2**: Same as SH_KNN but using kd-tree for quick nearest-neighbour lookup
9. **SH_KNN_3**: Same as SH_KNN but with $k = 5$
10. **SH_MAXENT1**: Maximum entropy distribution-based Shannon entropy estimator

$$H(\boldsymbol{Y}_{1:T}) = H(n) - \left[k_1\left(\frac{1}{T}\sum_{t=1}^{T} G_1(y'_t)\right)^2 + k_2\left(\frac{1}{T}\sum_{t=1}^{T} G_2(y'_t) - \sqrt{\frac{2}{\pi}}\right)^2\right] + log(\hat{\sigma}),$$

with

$$\hat{\sigma} = \hat{\sigma}(\boldsymbol{Y}_{1:T}) = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(y_t)^2},$$

$$y'_t = \frac{y_t}{\hat{\sigma}}, (t = 1, \ldots, T)$$

$$G_1(z) = ze^{\frac{-z^2}{2}},$$

$$G_2(z) = |z|,$$

$$k_1 = \frac{36}{8\sqrt{3} - 9},$$

$$k_2 = \frac{1}{2 - \frac{6}{\pi}},$$

11. **SH_MAXENT2**: Maximum entropy distribution-based Shannon entropy estimator, same as SH_MAXENT1 with the following changes:

$$G_2(z) = e^{\frac{-z^2}{2}},$$

$$k_2 = \frac{24}{16\sqrt{3} - 27},$$

12. **SH_SPACING_V**: Shannon entropy estimator using Vasicek's spacing method.

$$H(\boldsymbol{Y}_{1:T}) = \frac{1}{T}\sum_{t=1}^{T} log\left(\frac{T}{2m}[y_{(t+m)} - y_{(t-m)}]\right)$$

with T number of samples, the convention that $y_{(t)} := y_{(1)}$ if $t < 1$ and $y_{(t)} := y_{(T)}$ if $t > T$ and $m = \lfloor\sqrt{T}\rfloor$.

Bibliography

[1] Chen, W., Drton, M., Wang, Y.S.: On causal discovery with an equal-variance assumption. Biometrika **106**(4), 973–980 (2019)

[2] Daniusis, P., Janzing, D., Mooij, J., Zscheischler, J., Steudel, B., Zhang, K., Schoelkopf, B.: Inferring deterministic causal relations (2012), http://arxiv.org/abs/1203.3475

[3] Friedman, N., Nachman, I.: Gaussian process networks. CoRR **abs/1301.3857** (2013), http://arxiv.org/abs/1301.3857

[4] Hoyer, P., Janzing, D., Mooij, J.M., Peters, J., Schölkopf, B.: Nonlinear causal discovery with additive noise models. Advances in neural information processing systems **21**, 689–696 (2009)

[5] Hyvärinen, A., Smith, S.M.: Pairwise likelihood ratios for estimation of non-gaussian structural equation models. Journal of Machine Learning Research **14**(Jan), 111–152 (2013)

[6] Janzing, D., Hoyer, P.O., Schoelkopf, B.: Telling cause from effect based on high-dimensional observations (2009), http://arxiv.org/abs/0909.4386

[7] Janzing, D., Mooij, J., Zhang, K., Lemeire, J., Zscheischler, J., Daniušis, P., Steudel, B., Schölkopf, B.: Information-geometric approach to inferring causal directions. Artificial Intelligence **182**, 1–31 (2012)

[8] Judea, P.: Causality: models, reasoning, and inference. Cambridge University Press. ISBN 0 **521**(77362), 8 (2000)

[9] Kano, Y., Shimizu, S.: Causal inference using nonnormality. In: Proceedings of the international symposium on science of modeling, the 30th anniversary of the information criterion. pp. 261–270 (2003)

[10] Kap, B.: The effect of noise level on causal identification with additive noise models (2021), https://arxiv.org/abs/2108.11320

[11] Kohavi, R., Longbotham, R.: Online controlled experiments and a/b testing. Encyclopedia of machine learning and data mining **7**(8), 922–929 (2017)

[12] Kpotufe, S., Sgouritsa, E., Janzing, D., Schölkopf, B.: Consistency of causal inference under the additive noise model. In: International Conference on Machine Learning. pp. 478–486. PMLR (2014)

[13] Mooij, J., Janzing, D., Peters, J., Schölkopf, B.: Regression by dependence minimization and its application to causal inference in additive noise models. In: Proceedings of the 26th annual international conference on machine learning. pp. 745–752 (2009)

[14] Mooij, J.M., Janzing, D., Heskes, T., Schölkopf, B.: On causal discovery with cyclic additive noise models. In: Proceedings of the 24th International Conference on Neural Information Processing Systems. pp. 639–647 (2011)

[15] Mooij, J.M., Peters, J., Janzing, D., Zscheischler, J., Schölkopf, B.: Distinguishing cause from effect using observational data: methods and benchmarks. The Journal of Machine Learning Research **17**(1), 1103–1204 (2016)

[16] Nowzohour, C., Bühlmann, P.: Score-based causal learning in additive noise models. Statistics **50**(3), 471–485 (2016)

[17] Park, G.: Identifiability of additive noise models using conditional variances. Journal of Machine Learning Research **21**(75), 1–34 (2020)
[18] Park, G., Kim, Y.: Identifiability of gaussian structural equation models with homogeneous and heterogeneous error variances (2019), http://arxiv.org/abs/1901.10134
[19] Peters, J., Bühlmann, P.: Identifiability of Gaussian structural equation models with equal error variances. Biometrika **101**(1), 219–228 (11 2013). https://doi.org/10.1093/biomet/ast043, https://doi.org/10.1093/biomet/ast043
[20] Peters, J., Mooij, J., Janzing, D., Schölkopf, B.: Causal discovery with continuous additive noise models. Journal of Machine Learning Research **15**(1), 2009–2053 (2014)
[21] Rebane, G., Pearl, J.: The recovery of causal poly-trees from statistical data. CoRR **abs/1304.2736** (2013), http://arxiv.org/abs/1304.2736
[22] Sgouritsa, E., Janzing, D., Hennig, P., Schölkopf, B.: Inference of cause and effect with unsupervised inverse regression. In: Artificial intelligence and statistics. pp. 847–855. PMLR (2015)
[23] Shimizu, S.: Lingam: Non-gaussian methods for estimating causal structures. Behaviormetrika **41**(1), 65–98 (2014)
[24] Shimizu, S., Hoyer, P.O., Hyvärinen, A., Kerminen, A., Jordan, M.: A linear non-gaussian acyclic model for causal discovery. Journal of Machine Learning Research **7**(10) (2006)
[25] Shimizu, S., Hyvarinen, A., Kawahara, Y.: A direct method for estimating a causal ordering in a linear non-gaussian acyclic model (2014), http://arxiv.org/abs/1408.2038
[26] Silva, R., Scheines, R., Glymour, C., Spirtes, P., Chickering, D.M.: Learning the structure of linear latent variable models. Journal of Machine Learning Research **7**(2) (2006)
[27] Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search, vol. 81. Springer Science & Business Media (2012)
[28] Stegle, O., Janzing, D., Zhang, K., Mooij, J.M., Schölkopf, B.: Probabilistic latent variable models for distinguishing between cause and effect. Advances in neural information processing systems **23**, 1687–1695 (2010)
[29] Sun, X., Janzing, D., Schölkopf, B.: Causal inference by choosing graphs with most plausible markov kernels. In: Ninth International Symposium on Artificial Intelligence and Mathematics (AIMath 2006). pp. 1–11 (2006)
[30] Sun, X., Janzing, D., Schölkopf, B.: Causal reasoning by evaluating the complexity of conditional densities with kernel methods. Neurocomputing **71**(7-9), 1248–1256 (2008)
[31] Szabó, Z.: Information theoretical estimators toolbox. Journal of Machine Learning Research **15**, 283–287 (2014)
[32] Thase, M.E., Parikh, S.V., Rothschild, A.J., Dunlop, B.W., DeBattista, C., Conway, C.R., Forester, B.P., Mondimore, F.M., Shelton, R.C., Macaluso, M.: Impact of pharmacogenomics on clinical outcomes for patients taking medications with gene-drug interactions in a randomized controlled trial. The Journal of clinical psychiatry **80**(6), 0–0 (2019)

[33] Wright, S.: Correlation and causation. Journal of Agricultural Research **20**, 557–580 (1921)
[34] Young, S.W.: Improving library user experience with a/b testing: Principles and process. Weave: Journal of Library User Experience **1**(1) (2014)
[35] Zhang, K., Hyvarinen, A.: On the identifiability of the post-nonlinear causal model (2012), http://arxiv.org/abs/1205.2599

The effect of group roles on the development of online vaccination Twitter communities

T. Atsma[1], K. van der Zwet[1,2], and T. M. van Engers[1,2,3]

[1] Institute for Informatics, University of Amsterdam, Amsterdam, The Netherlands
[2] TNO, The Hague, The Netherlands
[3] Leibniz Institute, Amsterdam, The Netherlands

Abstract. This paper is focused on understanding the development of negative sentiment in online communities by studying their social structure and dynamics. A noticeable uprising of people has taken place against government vaccination programs and other measures during the recent COVID-19 crisis. Social networks play an important role in organising such activities. In this study, we formalize social roles in vaccination-related communities to understand the development of these communities. We defined three specific social roles: influentials, broadcasters, and commons. Using data from Twitter, we examined the possible effect of these roles upon the level of sentiment and activity on this social network. These effects were measured on a global scope, that is, on the entire data set, and a community scope, thus the effect users have on their community. It was concluded that the effect the influentials and commons have on the level of sentiment and activity of communities is similar. Only the broadcaster group differs from the other groups in some situations. Our research nuances the level of influence by "influentials" and influence by "common" people can be. Additionally, this work has shed more light on the limitations of a textual homogeneous data set, especially concerning topic modelling.

Keywords: Social roles · Community detection · Topic modelling · Sentiment analysis · Group development

1 Introduction

In our research, we study the sentiment and activity of online communities. Previous research shows that social systems are best characterized as complex adaptive systems (CAS), with internal and external relations, and interactions with their environment. In this study, we use a combination of quantitative methods including text mining, content analysis, and social network analysis. These methods enable us to empirically study social networks in a holistic manner. In this study, we focus on dynamics that occurred during the COVID-19.

Like other disasters, the SARS-CoV-2 virus, also known as the Coronavirus or COVID-19, has a tremendous global impact on people's lives. The disease and countermeasures against the spread of the disease cause distress in different

aspects of society on an economic, environmental, psychological, and sociological level [2]. More societal unrest emerges as more people become affected by these negative effects, which leads to a variety of social events [29]. The rise of such events is stimulated by the spread of disinformation [36, 20]. The likelihood that people become part of such an event depends on structural, social, and political context [25].

This unrest appears in different types of societal groups. Online networks have proven to be an effective tool for the mobilization, and organization of these groups [12, 33]. These groups are centred around a variety of topics, of which anti-vaccination discussions are a prominent example [17]. Additionally, empirical research identified that the vaccination hesitancy and intent to protest correlates with the development of negative sentiments in online discussions on these topics [9, 18].

The spread of vaccination-related disinformation is a problem and needs to be tackled. Epidemiologists have demonstrated that unvaccinated people are more susceptible to experience health issues, and obstruct the development of herd immunity [8]. Some claim vaccination is the only path to control a pandemic [28]. In order to better understand and battle this issue, one needs to understand how this information is spread within groups and how it potentially affects other groups, in particular in the context of online social media.

People can adopt different social roles within groups or communities [12, 11, 39]. Using data from Twitter, which is commonly used in this domain [5, 33], an attempt was made to answer the following research question: *What is the effect of social roles on the development of online vaccination-related Twitter communities?*.

Understanding the development of communities means understanding group dynamics over time. The dynamics, thus changes, that were studied are the changes in group activity and sentiment. To measure these changes we focus on the change in the number of messages and change of roles in communities. Additionally, we measure the sentiment level of the tweets. Analysing these dynamics should give us more insight into the group dynamics of online anti-vaccination related social media network groups during a pandemic. We, therefore, elicited how these groups develop over time, how information is spread, and who are potentially affected by this development, using a combination of network analysis and natural language processing techniques.

2 Background

During a pandemic, people appear to be more susceptible to stress, alleviated threat perception, and negative emotions such as fear and anxiety [2]. In turn, someone's judgement and choices are influenced by their emotional state during their decision making process as a reaction to their emotions [21]. Negative feelings, depending on a person's amount of self-efficacy, may produce behavioural changes or defensive responses. Additionally, it may guide people to information that reflects their beliefs, emotions, and behaviour. This information in turn will

affect their feelings, which influences their judgements and behaviour [2]. In extreme cases, these reactions may cause people to deliberately ignore important information. This phenomenon is referred to as an *infodemic* [30].

2.1 Vaccination groups

Network analysis has been applied to study the dynamics of vaccination-related discourse in online social groups [17]. The authors of [17] claim that features of nodes or clusters of nodes influence their network. They imply that the seemingly influential nodes of a network are not in dominant control over the system of the network. However, they state that their analysis of the relationships between network dynamics and node attributes can be improved through the analysis of other channels of interactions. Their formulae are approximations and the links between group members could be defined differently.

Similarly, in [41], it is confirmed that users with anti-vaccination sentiment tend to live in enclosed groups. This is a sign of the echo-chamber communication pattern. However, some pro-vaccination users take part in anti-vaccination discussions and are often the ones that initiate them. Moreover, it is mentioned that degree centrality is an insufficient measurement of actual influence. The results of Yuan et al. show that influential users do not always have a high in-degree and that the information of large organisations and accounts does not perturb the discourse of the target audience.

In online Facebook groups centred around vaccination, a small portion of its users is responsible for the largest portion of messages [6]. Moreover, in Twitter groups, it is shown that not all behavioural patterns are stimulated by online events, but also offline events [13].

2.2 Online communities

Before a community can be found, we have to formalise a definition for an online community. A variety of definitions of online communities exist that mostly differ depending on context [16]. Based on [16], the social roles will be seen as sociological factors that define the nature of the relationships being built between members of communities.

Community detection Community detection methodologies are designed to identify various communities in online networks. In [32], two noteworthy steps of the proposed methodology are feature selection and clustering. In this context, the features can be time intervals between tweets, common topics between users, and retweet activity. Features can also be the topics discussed in the interactions between network nodes. These topics can be extracted using topic modelling methods such as Latent Dirichlet Allocation (LDA) or Dirichlet Multinomial Mixture (DMM) [32, 34], of which the latter is the most promising for short texts [40]. After the feature selection step, the data can be clustered using models like K-Means or Order Statistics Local Optimization Method (OLSOM). These

clusters would form a type of community, depending on the selection of features. Thereafter, the quality of the clustering can be evaluated using metrics like the Silhouette Index, ARI, or NMI [1].

Nonetheless, there are algorithms specifically designed for community detection inside networks. Two well-known algorithms for community detection are the Louvain algorithm [4] and the Leiden algorithm [37]. The Louvain algorithm suffices in most situations. The Leiden algorithm solves the problems regarding the analysis of badly connected communities.

2.3 Social roles

Definition of a role The meaning of a definition of a social role depends on its context [10]. This means that the type of behaviour and actions associated with a role depending on the social context in which the role is situated. There are ways to conceptualize these definitions. In the context of Twitter, this behaviour can be found in how users interact with each other through messages, retweets, replies, and mentions, but also their language usage. Examples of node properties are the number of followers, friends on an individual level, and centrality metrics on a network level.

Role identification Social role identification has been studied in the field of sociology. In [39], the roles of *answer* and *discussion people* have been identified. These roles primarily respond to others or start threads of messages to which others respond. They should have replied or received a reply at least once. In a political context, the roles of *political influentials* and *opinion leaders* have been identified [7]. They are classified with metrics like network centrality and the clustering coefficient of a node. In [31], the roles of *social elites* are identified. These occupy a privileged position in their network. In other work, more related to social unrest and protests, four types of users are identified: *influentials*, *hidden influentials*, *broadcasters*, and *common users* [11]. Influentials are central users in the network and receive the highest number of messages. Hidden influentials do not take a prominent position in the network, but receive an above-average number of messages. Broadcasters have a central position in the network and send significantly more messages than they receive. Common users have relatively small audiences. These users are identified through generating two networks that capture who is following who and who is mentioned by who, respectively. In the case of the latter, information cascades are formed by tracking when a user broadcasts a message, and their direct neighbours respond within a short period.

Role behaviour It is shown that the behaviour of highly connected clusters in a network does not significantly affect the behaviour of the entire network. Highly influential members of a cluster are often not the ones behind cascades of influence. They strengthen the signal of others but do not take the initiative [27].

3 Method

3.1 Data

As mentioned in the introduction, the data that was used in this study was collected from the social media network Twitter. Fortunately, a data set containing tweets related to COVID-19 was already created and this data set is actively maintained, encompassing more than a billion tweets [19]. This data set contains a list of tweet ids.

Fig. 1. Flow of processing the data set of [19]

Initially, the data set proposed in [19] appeared to be the first viable source of data. The data set was processed in order to collect the desired information for this study. An overview of the process is displayed in figure 1. The original data set contains identifiers of tweets. The original tweet data can be retrieved using a process called *hydration* using a tool like *twarc* [4] or *hydrator* [5]. Different approaches were used to filter the tweets [3], on texts that contain vaccination-related keywords [15, 22], or on hashtags that relate to vaccination [24]. All three approaches led to a drastic decrease in the size of the data set. This gave the impression that the data set proposed by [19] is not suitable for performing vaccination-related analysis.

In consequence, a new data set was created using the Twitter Stream API. With the use of an academic Twitter developer account a script, inspired by a Twitter documentation page[6], was developed to perform a full archive search. The query included the same hashtags as proposed by [24]. The resulting data set has the same structure as the data set mentioned above and held around 493k tweets and were sent during the period of 2020-12-01 until 2021-04-01. The first of December was chosen as starting point as around this time the first official COVID-19 vaccinations were given.

In order to enable all computations with the data set, some data filtering was applied. First, irrelevant properties were removed. Additionally, rows in the data set with duplicate ids were removed. Finally, exploring the data clarified that many tweets were not related to vaccination but to the United States (US) elections, which were held in the same period. Therefore, the data set was filtered again but using a different approach as proposed by [3].

[4] https://github.com/DocNow/twarc
[5] https://github.com/DocNow/hydrator
[6] Source: https://github.com/twitterdev/Twitter-API-v2-sample-code/blob/master/Full-Archive-Search/full-archive-search.py

3.2 Content analysis

The data was preprocessed by removing punctuation, stop words, and special characters. Then the tweets were lower-cased, tokenised, and stemmed [23, 26, 35]. The effectiveness of stemming to improve polarity classification is not agreed upon by everyone. According to [23], its increase in performance is negligible, but not according to [35] of which the authors claim the performance dropped after removing the stemming operation. Therefore, this operation was kept in. Hereafter, the sentiment was classified using the Valence Aware Dictionary for sEntiment Reasoning (VADER) model [14]. VADER outperforms a Naive Bayes Analyzer and, according to the authors, operating as human classifiers. Additionally, topic analysis was performed. The algorithm of choice was Gibbs Sampling Dirichlet Mixture Model (GSDMM) [40]. It was preferred over the Latent Dirichlet Allocation model due to its poor performance because of the short-text nature of tweets. The performance of the GSDMM was evaluated by computing its topic coherence using the *UMass* topic coherence model. Topic coherence is used as a metric to qualitatively evaluate the semantics of learned topics. It can also be done intrinsically by computing a model's perplexity. The coherence scores ranged from -4.1 to -6.4. The UMass algorithm outputs negative numbers. The closer the value is to 0, the more coherent the topics are. As becomes clear, the coherence is less than optimal. Therefore, it was decided to exclude topics from the final results.

3.3 Community Detection

The Leiden model was selected due to its performance with regards to badly connected communities [37]. The goal is to apply the model to a network of interactions between twitter users. First, a set of nodes was made out of all unique users of the data set. Second, a set of edges was made out of unique interactions between users. Therefore, it was decided to add edges between nodes when the users represented by those nodes retweeted one another. Finally, running the Leiden model resulted in around 30k communities. Most of these communities have few members and did not seem too valuable. Therefore, it was decided to ignore all communities that hold less than 100 members. This left 84 distinct communities.

3.4 Role groups

Based on [12, 11] three role groups were defined: influentials, broadcasters, and commons. The role groups were identified using the following rules: the influentials are the users of which their number of followers is more than 10 times the average of their community. The broadcasters are the users of which their total number of tweets is more than 10 times the average. The common users are the top 20 users of which their number of followers is closest to the average. This means that some users are left out of the equation.

3.5 Sentiment effect

In order to systematically measure the effect a user role or role group has on the behaviour of a network or community, like the change in sentiment or activity, an equation is required to compute an effect score. For this purpose, the following variables were set up:

s = sentiment value of a tweet (between -1 and 1)
$G1$ = mean sentiment of a day
$G2$ = mean sentiment of the following day
A = G2 - G1 is the difference between the mean average of the first and second day. This represents a positive or negative change.
B = s - G1 is the difference between the actual sentiment value of a tweet and the mean sentiment value of the day the tweet was sent.

This gives the following equation:

$$f(s) = \frac{2 - |A - B|}{2} \tag{1}$$

This equation returns a floating-point number between 0 and 1. This score explains how similar the effect is to the difference in the sentiment of a single tweet and the average sentiment of the day the tweet was sent, measured by the change in the average sentiment between days. 2 is the maximum difference between the two extreme sentiment values. Therefore, if A and B are the same, thus achieving maximum similarity, the outcome is 1. If the two changes are maximally spread apart, the outcome is 0.

The goal is to find out if there is a difference between the effect values of different role groups, but also if role groups show different effect values on different scopes, that is, on the global and community scope. The global scope entails all messages whereas the community scope only entails the messages of a particular community. Per the scope, all variables as defined above were calculated and added as new features to the data set.

3.6 Activity effect

In addition to measuring the effect role groups have on the level of sentiment, a similar approach was used to measure the effect role groups have on the level of tweet activity. The following variables we extracted to enable this analysis:

$T1$ = the total number of tweets per day
$T2$ = the total number of tweets of the following day
$TD = T2 - T1$

Next, the data set was grouped into role groups to measure the average value of *TD* per role group. As with the sentiment effect, this process was done twice, once on a global level and once on a community level. The variables were added as new features to the data set. It is expected that on average the absolute value of *TD* will be larger for influentials as they tend to increase the pace of a trend. It is expected that the common users and broadcasters to not differ significantly as they send their messages within an already existing trend or non-existent trend. In short, the effect that is measured is if a certain role group has more or less impact on the general activity of users or communities.

4 Difference between roles

We first examine the results of our analysis on the sentiment of online communities. The means of the results of the sentiment effect function per role group were calculated. This was done on a global and community level. These results can be found in table 1.

Table 1. Means of the sentiment effect values per role group per scope

Scope	Influentials	Broadcasters	Commons
Global	0.8326	0.8308	0.8387
Community	0.8328	0.8205	0.8384

All values are close to 1, which means that the average level of the sentiment of all role groups is close to the average change in sentiment. Additionally, the difference between groups is low. Even the differences between the global and community level are low, except for the influentials. The global and community level difference of the influential role group appears to be more significant. In order to assess the actual significance of the differences between the levels and role groups, again, each community was iterated over. Per iteration, the mean of the sentiment effect function results of each member of a role group was computed and stored in an array. The mean was used instead of all the values as this could cause the sample size per community and per role group to become significantly unbalanced. Not all communities have the same number of members per role group and not every user has tweeted an equal number of times. In some cases, a community did not have any members of a specific role group. These missing values were imputed with the mean of their respective role group of all communities.

To test the significance of the difference between role groups, one has to decide what test to use. A well-known test is the Student's T-test. Though, this test assumes the two independent groups to come from a normally distributed population. In consequence, the final set of results were tested for normality of which almost no group passed.

A common alternative test is the Mann Whitney test, which is used to test whether the values of continuous variables of two independent groups differ significantly. It is an alternative to the unpaired t-test but does not assume the underlying distribution of the independent groups to come from a normally distributed population.

Finally, a different test is required to test the difference of the effect values of the same role group between different scopes. This is required because the two groups are the same population. The Wilcoxon test was used to test whether the distribution of the values of a role group differs significantly per scope. The results of these tests are shown in tables 2 and 3, respectively.

Table 2. Mann Whitney test for testing the difference of the effect between different role groups on the sentiment on different scopes with $p < \alpha = 0.05$ (True is reject h0, thus not the same distribution)

Scope	Group	Influentials	Broadcasters	Commons
Global	Influentials	-	False	False
	Broadcasters	False	-	False
	Commons	False	False	-
Community	Influentials	-	True	False
	Broadcasters	True	-	True
	Commons	False	True	-

Table 3. Wilcoxon test for testing the difference of the effect of the same role group on the sentiment between different scopes with $p < \alpha = 0.05$ (True is reject h0, thus not the same distribution)

Influentials	Broadcasters	Commons
False	True	False

The means of the groups are similar. The variances are roughly similar, except for the broadcasters which have a lower variance. This implies that broadcasters show more consistent behaviour. However, this could be because the broadcasters group is represented by a relatively high number of tweets as broadcasters are relatively more active. None of the groups was normally distributed, except for the broadcasters on the global scope. Additionally, on the global scope, none of the groups differed significantly from each other. On the community scope, the broadcasters differed significantly from the other groups. There is also a significant difference in the behaviour of the broadcasters on the global and community scope.

The same tests were run but with different data to test the effect of role groups on the change in tweet activity, that is, the number of tweets. On the global and community level, all communities were iterated over. Per iteration,

community, and role group the difference in the number of sent tweets per day was calculated of which the mean was taken and stored in an array. All missing values were imputed with the average of all averages. The results can be found in tables 4, 5, and 6.

Table 4. Means of the activity effect values per role group per scope

Scope	Influentials	Broadcasters	Commons
Global	-71.3298	15.4683	-210.7816
Community	-55.6423	-3.8584	-108.4216

Table 5. Mann Whitney test for testing the difference of the effect between different role groups on the activity on different scopes with $p < \alpha = 0.05$ (True is reject h0, thus not the same distribution)

Scope	Group	Influentials	Broadcasters	Commons
Global	Influentials	-	True	False
	Broadcasters	True	-	True
	Commons	False	True	-
Community	Influentials	-	True	False
	Broadcasters	True	-	True
	Commons	False	True	-

Table 6. Wilcoxon test for testing the difference of the effect of the same role group on the activity between different scopes with $p < \alpha = 0.05$ (True is reject h0, thus not the same distribution)

Influentials	Broadcasters	Commons
True	False	False

These results show a larger difference between role groups. On average, the broadcasters have the lowest negative effect on the change in tweet activity. Even more so on the global level, where on average the tweet activity increases the day after broadcasters have tweeted. On the community level, it drops slightly. The largest negative effects can be seen for the commons role group. Again, all results are not normally distributed. This means that the results of a role are most likely skewed towards the left or right, which implies that in most communities that role has a low or high impact on the activity, respectively. If this is not the case there are more differences between communities than expected. Next, similar to the results of the effect on the level of sentiment, only the broadcasters differ significantly from the other role groups. However, now also on a global level.

Interestingly, only the influentials differ significantly between the global and community level.

5 Discussion

In this study, we measured the effect of group roles on the development of sentiment and activity in a community. The results show that the effect of the common and influential groups is relatively similar and showed that all groups are similar on a global scope. This implies that the influence "common" users have is not as negligible and the influence "influential" users have is not as significant as one might think. In [5] it is mentioned that the number of followers does not necessarily correlate with influence. Similar to observations made by [11], where common users sometimes still manage to trigger certain reactions, due to their numbers. This idea is in line with the observations made in [27], in which it is said that influentials are not the drivers behind cascades of influence, but rather easily influenced users. Additionally, seemingly important vertices in a network are not always important to the behaviour of the system. In [5], it is shown that "average" users can become influential, especially when these users only concern themselves with a single topic. Likewise, this effect is seen in social movements and described as a bottom-up social force [38], where regular Twitter users promote their ideas through the social influence of celebrities with the use of hashtags. This pattern could not be shown due to the static nature of the interaction network. The properties of users are based on a static snapshot of the most recent state of those users. However, it is possible to measure the creation and removal of edges over time. This possibility should be explored further to verify the phenomenon of users that become increasingly influential. It would be interesting to find out how this phenomenon translates to a broader domain in order to get a different perspective on the formation and development of groups. Namely, that the difference in behaviour and influence of roles or functions of a group is not as apparent as expected. This perspective can aid organisations, advisory bodies, or governments in setting up campaigns. Instead of using resources to target influentials to set up a pro-vaccination campaign, it can be decided to spread messages among "common" people to set up an echo chamber effect. The same can be done to mitigate protests to control the level of social unrest in an area. However, it is still an open question to what extent these online patterns would translate to the real world.

The method chosen for classifying role groups prevents arbitrary overlap between users that should not fall into a specific role group. It also does not unnecessarily exclude users, resulting in a more accurate representation of a role group. However, it could cause the distribution of users to be out of balance. Additionally, it can lead to empty role groups. Furthermore, the commons role group was selected based on the top 20 most average users. This means that the below-average user role group was excluded.

The method also included an interaction network based on retweets alone. This was done for simplicity's sake and because retweets are the primary form of

interaction. However, there are several other forms of interaction that should be explored in future work. This work has also shed more light on the limitations of working with a textual homogeneous data set. In this context, homogeneous means the textual contents are biased towards a single topic and are based on similar contextual characteristics, that is, hashtags. This is especially true with regards to the poor performing topic models. Furthermore, it is expected that this also caused the activity effect values on a community level to be relatively negative. A possible explanation is that the used hashtags were viral and associated with popular topics [38]. This implies that many of the data points are recorded at the peak of interest and activity, after which the level of activity can only decline. Finally, we propose that the properties of the interaction network of users and communities should be further explored and explicated more explicitly in order to confirm our findings.

6 Conclusion

For this study, we used a method to measure the effect of behavioural user roles on the development of online vaccination Twitter communities. The analysis process consisted of processing Twitter data, detecting communities, defining and classifying user role groups, and measuring the effect these groups have on the average level of sentiment and user activity over time. These effects were measured on a global scope, that is, the entire data set, and on a community scope, which entails the effect users have on their community.

The means of the different role groups are similar and are close to 1. This means that, on average, the behaviour of each role group does not differ much. It also shows that all groups are relatively close to the general trend of the level of sentiment across the entire period that is recorded in the data set. Only the broadcaster group has a slightly lower effect on communities compared to the other roles and compared to its effect on the global scope. This difference is tested to be significant. The variances show that the influential and common groups are closely related. The broadcaster group has a much lower variance, which implies their behaviour is more consistent.

Between role groups, the differences of effect on the level of activity are larger than the differences between the effects on the level of sentiment. On average, almost all roles negatively affect global and community activity. This gives the impression that there are other factors at play. Only the broadcaster group has a positive effect on the global activity and a low negative effect on the community activity. This implies that when a broadcaster is active, the global trend sees a positive change on the following active day. The opposite is true for the influential and common groups. Again, the effects are not normally distributed and the differences between the broadcaster group and the other groups are significant. A noteworthy mention is that the effect of the influential group on the global activity and the effect on the community activity differ significantly. Both results imply that the broadcaster group shows the most unique behaviour. It also implies that the broadcaster group seemingly has the

most potential to stimulate the activity of others. In general, it can be concluded that except for the broadcaster group, the influential and common role groups appear to be similar in terms of their effect on the development of communities. Except for the broadcaster group, the influential and common role groups appear to be similar in terms of their effect on the development of communities.

In conclusion, our study contributes to our broader interest to understand the mechanisms behind insurgent behaviour. While some insurgent behaviour may be fueled by social media, physical social networks also consist of people adopting certain social roles with different impacts on group behaviour. To what extent the mechanisms found in online groups could be translated to such physical groups is still an open question. Further research is needed to further explore the translation of online patterns to the offline world, but also to explore the inclusion of below-average users in order to find hidden influencers, to research the design of a more comprehensive and inclusive role classification methodology [5, 11], to explore if community size affects the results, and to explore the implications of homogeneous textual data.

References

1. Baarsch, J., Celebi, M.E.: Investigation of internal validity measures for k-means clustering. Lecture Notes in Engineering and Computer Science **2195**, 471–476 (3 2012)
2. Bavel, J.J.V.: Using social and behavioural science to support covid-19 pandemic response. Nature Human Behaviour **4**, 460–471 (5 2020). https://doi.org/10.1038/s41562-020-0884-z
3. Bello-Orgaz, G., Hernandez-Castro, J., Camacho, D.: Detecting discussion communities on vaccination in twitter. Future Generation Computer Systems **66**, 125–136 (1 2017). https://doi.org/10.1016/j.future.2016.06.032
4. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**, 10008 (10 2008), https://iopscience.iop.org/article/10.1088/1742-5468/2008/10/P10008
5. Cha, M., Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.P.: Measuring user influence in twitter: The million follower fallacy. In ICWSM '10: Proceedings of international AAAI conference on weblogs and social (2010), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.167.192
6. Chiou, L., Tucker, C.: Fake news and advertising on social media: A study of the anti-vaccination movement (11 2018). https://doi.org/10.3386/w25223, http://www.nber.org/papers/w25223.pdf
7. Dubois, E., Gaffney, D.: The multiple facets of influence: Identifying political influentials and opinion leaders on twitter. American Behavioral Scientist **58**, 1260–1277 (9 2014), https://doi.org/10.1177/0002764214527088
8. Fefferman, N.H., Naumova, E.N.: Dangers of vaccine refusal near the herd immunity threshold: A modelling study. The Lancet Infectious Diseases **15**, 922–926 (8 2015). https://doi.org/10.1016/S1473-3099(15)00053-5
9. Freeman, D., Waite, F., Rosebrock, L., Petit, A., Causier, C., East, A., Jenner, L., Teale, A.L., Carr, L., Mulhall, S., Bold, E., Lambe, S.: Coronavirus conspiracy

beliefs, mistrust, and compliance with government guidelines in england. Psychological Medicine pp. 1–13 (2020), https://doi.org/10.1017/S0033291720001890
10. Gleave, E., Welser, H.T., Lento, T.M., Smith, M.A.: A conceptual and operational definition of 'social role' in online community. 2009 42nd Hawaii International Conference on System Sciences pp. 1–11 (1 2009). https://doi.org/10.1109/HICSS.2009.6
11. González-Bailón, S., Borge-Holthoefer, J., Moreno, Y.: Broadcasters and hidden influentials in online protest diffusion. American Behavioral Scientist **57**, 943–965 (7 2013), https://doi.org/10.1177/0002764213479371, publisher: SAGE Publications Inc
12. González-Bailón, S., Borge-Holthoefer, J., Rivero, A., Moreno, Y.: The dynamics of protest recruitment through an online network. Scientific Reports **1**, 1–7 (12 2011). https://doi.org/10.1038/srep00197
13. Gunaratne, K., Coomes, E.A., Haghbayan, H.: Temporal trends in anti-vaccine discourse on twitter. Vaccine **37**, 4867–4871 (8 2019). https://doi.org/10.1016/j.vaccine.2019.06.086
14. Hutto, C.J., Gilbert, E.: Vader: A parsimonious rule-based model for sentiment analysis of social media text (2014), http://sentic.net/
15. Immunology.org: Celebrate vaccines: key words in vaccine immunology — british society for immunology (2020), https://www.immunology.org/news/celebrate-vaccines-key-words-in-vaccine-immunology
16. Iriberri, A., Leroy, G.: A life-cycle perspective on online community success. ACM Computing Surveys **41**, 1–29 (2 2009), https://dl.acm.org/doi/10.1145/1459352.1459356
17. Johnson, N.F., Velásquez, N., Restrepo, N.J., Leahy, R., Gabriel, N., Oud, S.E., Zheng, M., Manrique, P., Wuchty, S., Lupu, Y.: The online competition between pro- and anti-vaccination views. Nature **582**, 230–233 (6 2020), https://www.nature.com/articles/s41586-020-2281-1
18. Jolley, D., Douglas, K.M.: The effects of anti-vaccine conspiracy theories on vaccination intentions. PLoS ONE **9**, 89177 (2 2014). https://doi.org/10.1371/journal.pone.0089177, www.plosone.org
19. Lamsal, R.: Design and analysis of a large-scale covid-19 tweets dataset. Applied Intelligence pp. 1–15 (11 2020), https://doi.org/10.1007/s10489-020-02029-z
20. van der Linden, S., Roozenbeek, J., Compton, J.: Inoculating against fake news about covid-19. Frontiers in Psychology **11**, 2928 (10 2020). https://doi.org/10.3389/fpsyg.2020.566790, www.getbadnews.com
21. Loewenstein, G.F., Hsee, C.K., Weber, E.U., Welch, N.: Risk as feelings. Psychological Bulletin **127**, 267–286 (2001). https://doi.org/10.1037/0033-2909.127.2.267
22. Love, B., Himelboim, I., Holton, A., Stewart, K.: Twitter as a source of vaccination information: Content drivers and what they are saying. American Journal of Infection Control **41**, 568–570 (6 2013). https://doi.org/10.1016/j.ajic.2012.10.016
23. Mejova, Y., Srinivasan, P.: Exploring feature definition and selection for sentiment classifiers (7 2011), http://wordnet.princeton.edu/
24. Memon, S.A., Tyagi, A., Mortensen, D.R., Carley, K.M.: Characterizing sociolinguistic variation in the competing vaccination communities. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **12268 LNCS**, 118–129 (10 2020)
25. Newburn, T.: The causes and consequences of urban riot and unrest. Annual Review of Criminology **4**, 53–73 (2021), https://doi.org/10.1146/annurev-criminol-061020-124931

26. Porter, M.F.: An algorithm for suffix stripping (3 1980). https://doi.org/10.1108/eb046814
27. Quax, R., Apolloni, A., Sloot, P.M.: The diminishing role of hubs in dynamical processes on complex networks. Journal of the Royal Society Interface **10** (11 2013), http://dx.doi.org/10.1098/rsif.2013.0568
28. Rasmussen, A.L.: Vaccination is the only acceptable path to herd immunity. Med **1**, 21–23 (12 2020). https://doi.org/10.1016/j.medj.2020.12.004
29. Reicher, S., Stott, C.: On order and disorder during the covid-19 pandemic. British Journal of Social Psychology **59**, 694–702 (2020), https://bpspsychub.onlinelibrary.wiley.com/doi/abs/10.1111/bjso.12398
30. Shahi, G.K., Dirkson, A., Majchrzak, T.A.: An exploratory study of covid-19 misinformation on twitter. Online Social Networks and Media **22**, 100104 (3 2021). https://doi.org/10.1016/j.osnem.2020.100104
31. Siegel, D.A.: Social networks and collective action. American Journal of Political Science **53**, 122–138 (2009), https://www.jstor.org/stable/25193871
32. Silva, W., Ádamo Santana, Lobato, F., Pinheiro, M.: A methodology for community detection in twitter. Proceedings of the International Conference on Web Intelligence pp. 1006–1009 (8 2017), https://doi.org/10.1145/3106426.3117760
33. Steinert-Threlkeld, Z.C., Mocanu, D., Vespignani, A., Fowler, J.: Online social networks and offline protest. EPJ Data Science **4**, 1–9 (12 2015), https://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-015-0056-y, number: 1 Publisher: SpringerOpen
34. Surian, D., Nguyen, D.Q., Kennedy, G., Johnson, M., Coiera, E., Dunn, A.G.: Characterizing twitter discussions about hpv vaccines using topic modeling and community detection. Journal of Medical Internet Research **18**, e232 (2016), https://www.jmir.org/2016/8/e232/
35. Symeonidis, S., Effrosynidis, D., Arampatzis, A.: A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. Expert Systems with Applications **110**, 298–310 (11 2018). https://doi.org/10.1016/j.eswa.2018.06.022
36. Thomas, B.P.C.: Worldwide protests in 2020: A year in review. Carnegie Endowment for International Peace (2020), https://carnegieendowment.org/2020/12/21/worldwide-protests-in-2020-year-in-review-pub-83445
37. Traag, V.A., Waltman, L., van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. Scientific Reports 2019 9:1 **9**, 1–12 (3 2019), https://www.nature.com/articles/s41598-019-41695-z
38. Wang, R., Liu, W., Gao, S.: Hashtags and information virality in networked social movement: Examining hashtag co-occurrence patterns. Online Information Review **40**, 850–866 (2016). https://doi.org/10.1108/OIR-12-2015-0378
39. Welser, H.T., Gleave, E., Fisher, D., Smith, M.: Visualizing the signatures of social roles in online discussion groups p. 32 (2007)
40. Yin, J., Wang, J.: A dirichlet multinomial mixture model-based approach for short text clustering. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp. 233–242 (8 2014), https://dl.acm.org/doi/10.1145/2623330.2623715
41. Yuan, X., Crooks, A.T.: Examining online vaccination discussion and communities in twitter. Proceedings of the 9th International Conference on Social Media and Society (2018), https://doi.org/10.1145/3217804.3217912

A study of BERT's processing of negations to determine sentiment

Giorgia Nidia Carranza Tejada[1], Johannes C. Scholtes[1,2], and Gerasimos Spanakis[1][0000−0002−0799−0241]

[1] Maastricht University. Faculty of Science and Engineering, Department of Data Science and Knowledge Engineering
g.carranzatejada@student.maastrichtuniversity.nl,
jerry.spanakis@maastrichtuniversity.nl
[2] iPro Tech LLC-ZyLAB Technologies
j.scholtes@maastrichtuniversity.nl

Abstract. The question considered in this paper is: can BERT effectively distinguish the meaning of the following two sentences: '*BERT is capable of understanding negations*' and '*BERT is not capable of understanding negations*'? This work aims to fulfill the gap in the knowledge about BERT's capacity to handle negations. The specific task under examination is sentiment analysis, where erroneous understanding of negations directly affects the model's performance by wrongly switching polarity of the detected sentiments. In order to determine what BERT 'understands' from negated text, a model was trained and tested by using adversarial conditions. With four distinct configurations, handling negations was studied by interchanging negated sentences during training and testing. The results exposed that in three out of four cases, the BERT's propensity to deal with negations by memorizing information in the large number of connections used by the model, instead of truly understanding the linguistic mechanism of negations. In the remaining case, the model's performance suggested taking decisions based on random features without exposing clear reasoning. Based on these insights, best-practice methods for training BERT to deal better with negations in sentiment analysis can be formulated.

1 Introduction

An area widely investigated in text mining, is sentiment analysis. Sentiment analysis studies techniques to identify and examine human sentiments towards different experiences and interests. In general, the sentiments expressed in a text are positive, negative, or neutral [11].

In order to obtain the correct classification of a sentence, it is essential to handle negations correctly. Not doing so, will impact the polarity of the sentiment, resulting in wrong classifications. An example is the following positive sample: *"this is a good film"*, if it is negated, this sentence expresses a negative opinion: *"this is not a good film"*. So, not dealing correctly with the negation, will change the polarity in the exact opposite direction.

One of the State-of-the-Art (SotA) models to detect and classify sentiments, is BERT (Bidirectional Encoder Representations from Transformer) [14]. Despite the high quality of classification, a detailed error analysis indicates that the majority of misclassifications comes from erroneous handling of negations: a percentage of 66% in comparison to the other error categories. Figure 1 shows the distribution among the classes of errors indicated by [8].

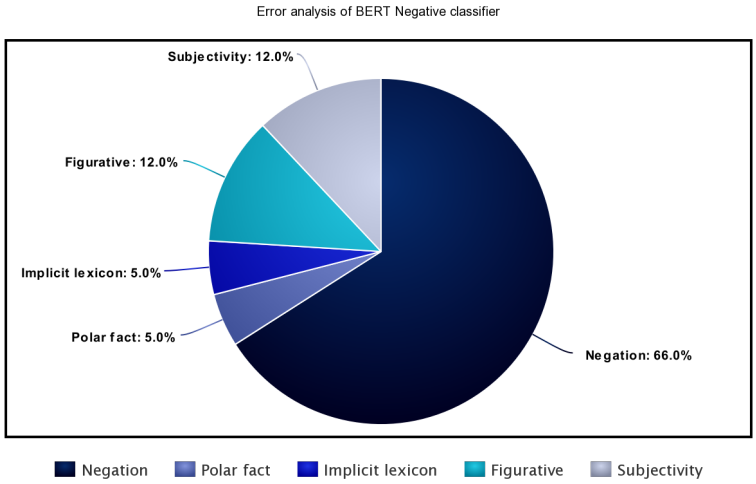


Fig. 1: Distribution of BERT's error analysis for the sentiment analysis task

So, although BERT holds the state-of-the-art result for several Natural Language Processing (NLP) tasks, Figure 1 indicates that BERT is not really capable of handling negations. Given the fact that 66% of errors in the sentiment-analysis task originate from wrongly handling negations, the research in this paper focuses on better understanding BERT's negation-handling mechanisms, in order address these errors and increase the overall result of the sentiment-analysis task.

There are a number of reasons why we believe there is a deeper issue at hand with BERT's negation handling skills: (i) the mechanism behind Word Embeddings assign similar encodings to words used frequently in the same context. In other experiments conucted by [13], it has been observed that words of completely different polarity get very similar encodings (e.g. *good* versus *bad* or *happy* versus *unhappy*). This confuses the classification in tasks such as sentiment analysis, where the polarity is more important than in other tasks such as machine translation. (ii) Due to the enormous amount of trainable connections, BERT has tremendous memory skills. But memorizing is something very different from inferencing the polarity of (double) negations. (iii) BERT's attention mechanism seems to be based on the presence of certain specific cue words it memorizes, thereby missing words relevant for negations such as *not*.

Our work extends prior studies, examining BERT's behavior in an adversarial condition during training and testing. By investigating BERT's predictions based on certain training data, it should be possible to better understand in certain situations where BERT's errors dealing with negations originate from, so they can be better addressed in future sentiment analysis models.

2 Related Work

As stated before, notable advancements in various NLP tasks were produced after the introduction of BERT in 2018. Despite these impressive results, there has also been interest what BERT is not capable of, especially by the computational-linguistic community. For example, [2] analyzed linguistic errors, the problem derived from the commonsense, pragmatic inference, and negation. These experiments showed that BERT failed to adjust to negated statements: the predictions persisted unaltered after the insertion of the negation. See Figure 2 for a number of examples of such wrong prediction. From these, it is completely clear that BERT is completely ignoring the negation!

Context	$\text{BERT}_{\text{LARGE}}$ predictions
A robin is a ____	*bird, robin, person, hunter, pigeon*
A daisy is a ____	*daisy, rose, flower, berry, tree*
A hammer is a ____	*hammer, tool, weapon, nail, device*
A hammer is an ____	*object, instrument, axe, implement, explosive*
A robin is not a ____	*robin, bird, penguin, man, fly*
A daisy is not a ____	*daisy, rose, flower, lily, cherry*
A hammer is not a ____	*hammer, weapon, tool, gun, rock*
A hammer is not an ____	*object, instrument, axe, animal, artifact*

Table 13: $\text{BERT}_{\text{LARGE}}$ top word predictions for selected NEG-136-SIMP sentences

Fig. 2: Table from "[What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models.]", by Ettinger, 2020, *Transactions of the Association for Computational Linguistic*

This work was a direct extension of [3], which focused on analyzing BERT's syntactic abilities by supplying an entire sentence to BERT, while masking out the single focus verb.

Another study proposed by [7] proved the deterioration of BERT performance when denials were added in claims for argument comprehension tasks.

[5] investigated the effect of the negation on the question-answering task by applying the masked language model. The research concluded that BERT's can learn predictions based on exact phrases shown during training, whereas it

poorly generalizes over a test set that contains phrases it did not see during training.

Also, [4] focused on understanding how a Pre-Trained Language model (hereinafter PLM) like BERT learns factual knowledge from the training data. During the symbolic reasoning analysis, e.g. the ability of a PLM to deduce information not shown during the pre-training, a rule explicitly investigated was negation. The work assumed that the general concept of denial was not understood while co-occurrence is used to acquire antonym negation. Kassner's study involved [9]'s prior work, which connects BERT's prediction to the knowledge-base and (lack of actual) inferencing capabilities.

Other studies into the relation of BERT and negation handling can be found in [6], which focused on employing BERT to detect the denial and delimiting its scope, and [15], which analyzed a plausible relation between a negation cue and its scope in the attention heads. Both confirmed lack of actual knowledge of the negations by measuring significant inconsistencies in the average negation detection.

Our approach is similar to the work of Ettinger and Goldberg, as we focus more on BERT's word prediction capability, specifically on the sentiment analysis task. Our work differs from Kassner's, which focuses more on the detection of negations while we examine the effect of negations on sentiment prediction.

3 Methodology

BERT's strength comes from the simplicity of adjusting the original pre-trained model configuration for a specific task by fine-tuning the model, thereby taking advantage of transfer learning. Instead of having to learn language from scratch, BERT has basic linguistic skills resulting from being exposed to many billions of words in their linguistic context. But, for reasons not well understood, this ability does not apply to negation handling, which obviously is very important in sentiment analysis. The question we ask ourselves: is BERT just memorizing training data using its large number of parameters, or does it actually "understands" negations?

From initial observations, our hypothesis is that BERT tends to memorize. To verify the validity of this assumption, BERT's ability to deal with negations will be studied in adverse training and testing conditions. The examination will follow a similar approach as Ettinger's work (see figure 2 for examples), where the knowledge of BERT is questioned by adding negations to testing sentences that the model did not see before. So, if the model is indeed only memorizing, the prediction will remain unchanged after these perturbations.

To begin with, two BERT classifiers are trained with labeled sentences from the SemEval 2017 task 4a ([11]) and SST5 ([12]) data set. One binary positive classifier and one binary negative classifier ([1]). Subsequently, classification is tested on negated sentences for each class. From the examples in figure 3, it can be observed that in both cases BERT predicts the original sentiment and not the negated sentiment.

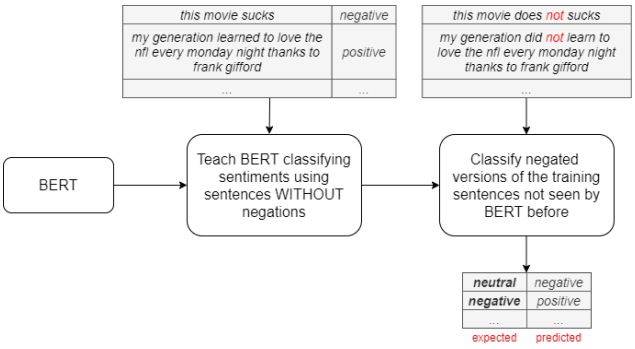


Fig. 3: BERT's misclassifications after the addition of negations.

In subsequent experiments, four different approaches were used to confirm that BERT actually memorizes. First, in model M1 we will include only sentences without negations in the training set. The test set then contains negated versions of these sentences. In model M2 , we take the reverse approach, where the training set only includes negated sentences and the test set then contains non-negated versions of these sentences. Then, model M3 will include in the training data set randomly one of the two versions of the sentence. The inverted sentence of each will be considered in the testing. So, in M3 the system will be exposed to both negated and non-negated sentences, but in the testing sentences, the negations will be opposite from the training sentences. Finally, model M4. includes for every sentence both the negated and the non-negated version. We then test on a validation set, containing sentences not seen before by the model during training.

The configurations selected for the training and testing are briefly explained in table 1

Model	Negated	Not Negated
M1	none	all samples
M2	all samples	none
M3	either	either
M4	all samples	all samples

(a) The data set employed during the training.

Model	Negated	Not Negated
M1	all samples	none
M2	none	all samples
M3	remaining	remaining
M4	test samples	test samples

(b) The data set employed for the testing of the model.

Table 1: The tables represent the configuration examined to verify the performance of BERT towards the negations.

If experiment M1 and M2 result in BERT predicting the original sentiment from training instead of the negated sentences, and if BERT shows inconsistent

behavior for experiment M3, then it is clear that BERT is actually memorizing on cue words instead of really understanding the linguistic operation of negation.
In order to demonstrate this more convincingly, the behavior of connotations such as *can not* versus *cannot* versus *can't* will be investigated. Because, if BERT can deal with *cannot* and *can't* but not with *can not*, then the case for memorization of cue words is even stronger.
This, and the influence of specific negations words such as *not* is studied by using through the Local Interpretable Model-Agnostic Explanations (hereinafter LIME) approach. LIME is a technique employed to explain a prediction of any black-box machine learning model by presenting qualitative connections between the instance's components and the model's prediction [10]. The methodology proposed by LIME consisted on performed a local fidelity analysis by initially altering the original data point before being fed into the model. Then, the importance of each feature is represented by the change in the predictions obtained.

The interpretation obtained is not the faithful representation of the entire model but is reliable locally, which depends on the performance obtained in the proximity of the sample examined. Additionally, LIME guarantees an interpretable representation by applying bag-of-word when it is needed for text classification.

In the example proposed by the paper, the technique helps to understand the eventual cue words learned by the model to determine the class of the text. The explanations evidenced an issue of the classifier related to the data set selected. The same approach will be used during this examination to provide more insights into the impact of negation words on the sentiment predictions.

4 Experiments

For the experiments, a sentiment classifier was built using the Transformers library by HuggingFace supported by the PyTorch Machine Learning framework. The number of epochs employed in our experiment is equal to 2, the number of batches is set to 32, and the optimizer selected was AdamW. The model loaded from the Transformers library, represented only the hidden layer of the input tokens. The output is passed through linear transformation.

Since the main purpose of the baseline([1]) was to detect either positive or negative sentiment in a neutral context, it was considered to employ a binary classification using the one-vs-rest technique. The choice derived from the overrepresentation of the neutral class influenced the multi-class model performance to assign an incorrect neutral label in most cases. So, the sentiment analysis system consists of two binary classifiers: the positive classifier trained as positive against either negative or neutral and the negative classifier trained as negative against either positive or neutral.

To sum up, each sentence to be evaluated is fed into both the classifiers as input. Then, the outputs, which correspond to the outcomes of the binary classifiers, identify the sentiment in the text.

Starting data sets for our analysis are the concatenation of SemEval-2017 and SST5, as those used for the baseline model. The first was provided by task 4a of the International Workshop on Semantic Evaluation (SemEval) 2017, Sentiment Analysis in Twitter1. CrowdFlower or Mechanical Turk realized the annotations for each tweet [11]. Besides, the labels complied with the three sentiment categories previously nominated, and they are distributed as follows: 34% positive, 16% negative, and 50% neutral.

Then, the data set SST5, published on [12], is a fine-grained sentiment data set containing five different labels: 0 (very negative), 1 (negative), 2 (neutral), 3 (positive), 4 (very positive). Because our experiments consider only three labels (negative, neutral, and positive), the labels *very positive* and *very negative* were included in respectively *positive* and *negative*. The overall balance of the labels in our combined data set is defined as follows: 42% positive, 39% negative, and 19% neutral.

Name	Size	Positive	Negative	Neutral
SemEval 2017 task 4a	20631	34%	16%	50%
SST5	8544	42%	39%	19%

Table 2: The size and labels' distribution of the sentiment data sets.

The original data set presented an unbalance among the two categories: negated cases were only 22% of the entire collection. To balance, the data set was augmented through transformation functions as defined in table 3. This then resulted in a more balanced distribution, with 49.4% negated sentences and 50.6% not negated ones, and in different data sets for experiments M1, M2, M3. and M4. For training 90% of the modified data is used. For testing the remaining 10%.

Transformation function	**Original sentence**	**Modified sentence**	**Created samples**
Addition of the negation	Paul Bettany is cool	Paul Bettany is *not* cool	15792
Removal of the negation	he 's *not* good with people	he 's good with people 1	8569

Table 3: The table defined the transformation functions used in the examination to alter the original data.

5 Results and Discussion

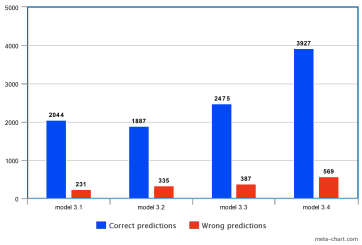
Table 4 collects the outcomes of the binary classifiers obtained from the test on the original sentences, which are the samples following the configurations established for the training data. So this table represents how well the classifier works on similar sentences to the training data. While table 5 assembles the performances on negated versions these sentences.

	Positive			Negative		
Model	**Precision**	**Recall**	**F_measure**	**Precision**	**Recall**	**F_measure**
M1	0.75	0.79	0.77	0.73	0.76	0.74
M2	0.93	0.96	0.94	0.87	0.94	0.90
M3	0.90	0.92	0.91	0.82	0.84	0.83
M4	0.88	0.90	0.89	0.84	0.89	0.86

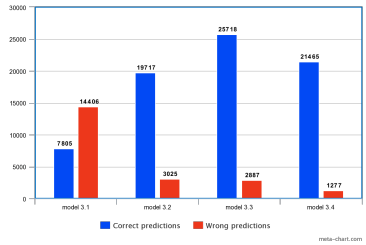
Table 4: Original sentences

	Positive			Negative		
Model	**Precision**	**Recall**	**F_measure**	**Precision**	**Recall**	**F_measure**
M1	0.89	0.28	0.42	0.75	0.20	0.32
M2	0.51	0.93	0.66	0.66	0.63	0.65
M3	0.83	0.86	0.85	0.79	0.79	0.79
M4	0.88	0.89	0.89	0.84	0.89	0.86

Table 5: Negated sentences

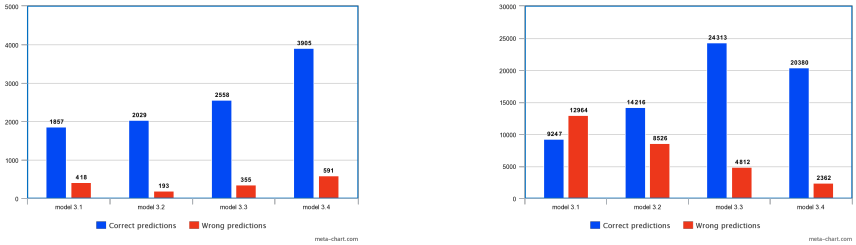


(a) Original sentences



(b) Negated sentences

Fig. 4: Distribution of the correct and wrong predictions for the **negative** classifier (model M1 to M4)



(a) Original sentences (b) Negated sentences

Fig. 5: Distribution of the correct and wrong predictions for the **positive** classifier (model M1 to M4)

5.1 Model M1, trained with non-negated sentences and tested on negated versions of these sentences.

Model M1 was trained by not showing negations during training. Then, to question BERT's capabilities to deal with negations, training sentences were negated and used as test.

As expected, BERT is not able to handle the negations in the test sentences. Indeed, the performance drops significantly on the negated sentences compared to the non-negated ones, as can be observed from the high error in the left columns of figure 4b and 5b.
Furthermore, LIME was used to examine the wrongly predicted sentence: '*about to go shopping again tomorrow bc the dress I got for jason aldean is not cute.*' in more detail.

The respectively original sentence was: '*about to go shopping again tomorrow bc the dress I got for jason aldean is cute*', and holds a positive sentiment. When negated, the prediction should be reversed and classified as negative, but the negative classifier failed to identify the sentiment. Figure 6 provides more insight how the prediction was made by using LIME. The LIME's process to represent the feature's impact in the prediction evidenced that the negation cue and its scope had been recognized by BERT and correctly attributed to the negative class. However, the significance of these words was not enough to invert the overall label since the distance of the class none was still considerable compared to the negative class, which was the correct prediction.

5.2 Model M2, trained with negated sentences and tested on non-negated versions of these sentences.

Model M1 is the reverse of model M2: training included only negated sentences, where testing was done with non-negated versions of the training sentences.

In this case, the performance in the non-negated sentences did not decrease as drastically as in model M1. Additionally, it was observed that the most common error originated from issues dealing with subjectivity/objectivity and figurative

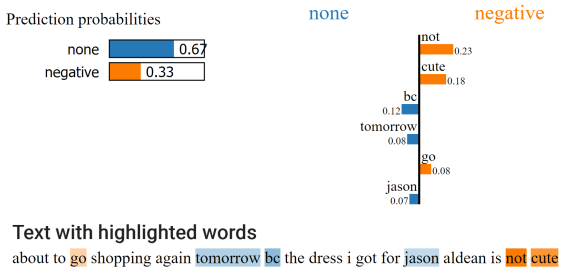


Fig. 6: The result of LIME in the sentence '*about to go shopping again tomorrow bc the dress I got for jason aldean is not cute.*' for the negative classifier.

language. Therefore, the result seems to indicate that BERT was actually learning something about dealing with negations.
However, further analyzing the model in more details with respect to connotations such as *cannot* or *can't* versus *can not*, it becomes clear that BERT's decisions are actually still based on memorization. Considering that the training data included one version of a negated verb, either extended (*is not*) or contracted (*isn't*), then the prediction should not be influenced if the negated verb is replaced by the contracted connotation in the test data. For example, table 6 represents the situation including the same sentence: firstly with the verb negated by using a separate *is not* and then by using the contracted connotation: *isn't*. The prediction of the system should be equal, but in our case, it changed for the contracted connotation. So, evidently, BERT did not "understand" negations but based it's decision on memorization.

text	label	prediction
solondz is so intent on hammering home his message that he does forget to make it entertaining	negative	-
solondz is not so intent on hammering home his message that he does not forget to make it entertaining	positive	negative
solondz isn't so intent on hammering home his message that he doesn't forget to make it entertaining	positive	positive

Table 6: An example of the classification of the same sentence but changing the type of negated verb: firstly the extended, then the connotation.

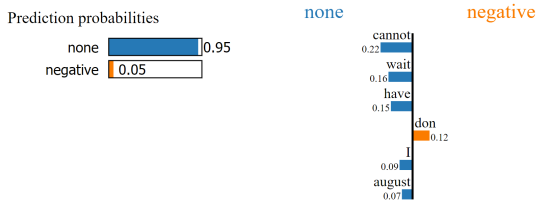
5.3 Model M3, Trained on either negated or non-negated sentences and tested on the inverse for each sentence.

The third model examined was M3, which was trained on the data set composed of a random selection of one version of a sentence: either the negated or the non-negated version. Then, testing was done on the inverse of the train sentences.

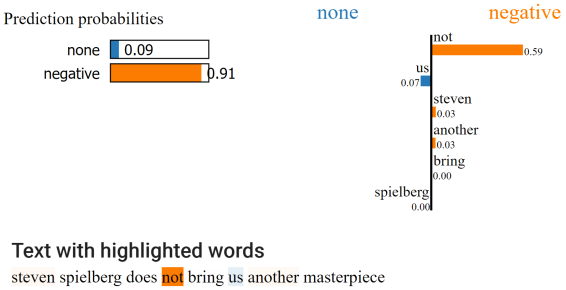
So, for the negated sentences, a non-negated one was used and visa-versa. In this case, there were more correctly predicted sentences than in model M1 and M2.

Even though, by examining in-depth the correctly or wrongly classified sentences, it was not possible to deduct any particular pattern why some sentences were classified correctly or wrongly. Consequently, the model seemed to take decisions based on random features, which were not clearly understandable from either the predictions or through the deployment of LIME.

In particular, the examination aimed to identify a common pattern, similar to the previous case, where the negation's impact was repeatedly neglected for the prediction, or the use of connotations underlined an inaccurate behavior of the model. In this state, no anomalies were detected from the employment of connotations, and as well, the negations were sometimes correctly classified and sometimes not, without establishing a frequent behavior. For instance, the latter case was represented in the results obtained by LIME in two sentences. First, an exact classification correctly identified and handled the negation. Then, an erroneous prediction was determined from a sentence with a double negation. The initial negation did not alter the sentence, but the second did by reversing it. Although the second negation cue was identified and attributed to a negative class, the impact on the total prediction was not decisive.



(a) Wrongly predicted sentence.



(b) Correct predicted sentence.

Fig. 7: LIME results on one correct and one wrong prediction of model M3.

5.4 Model M4, Train with both negated and non-negated versions of each sentence, and test on an external validations set

Finally, the last model, M4, obtained the highest result in the testing on negated sentences. Additionally, it achieved the smaller variation in the precision, recall, and f-measure between the original and negated results. This performance was strictly connected to the configuration adopted by the model since the testing was on a direct subset of the training data. Therefore, from this last experiment, one could derive that BERT capabilities to deal with negations are based on memorization.

6 Conclusion

The goal of this research was to contribute to a better understanding of the underlying mechanisms of BERT's negation handling. Therefore BERT's behavior was investigated by testing adversarial sentences in sentiment analysis, where the effects of wrong negation handling has much more impact than in other linguistic tasks.

This research indicates that BERT's handling of negations is more based on it's tremendous ability to memorize rather than "understanding" the negation.

Notably, BERT was unable to learn how to properly deal with denial when trained only on denied or non-denied sentences. Indeed, in the first case, the model's predictions turned out to be random, as evidenced by the connotation example, while in the second case, the model ignored the negation. The optimum results were achieved by the last model, where each sample included both the negated and non-negated version of all sentences. This configuration was able to take full advantage of BERT's memorization capabilities and resulted in the highest f1 scores.

In conclusion, for future realization of sentiment analysis systems where negations will be properly addressed, it is recommended that the data sets contain a proportioned distribution of negated and non-negated cases for each sentence. Additionally, it is also suggested that future data sets for sentiment analysis competitions (e.g. the ones used in SemEval and SST5), spend more attention to dealing with negations, as this is a major source of errors in the real-world.

7 Acknowledgements

The authors wish to thank ZyLAB Technologies BV in Amsterdam, the Netherlands for providing the funding and resources to realize this research. ZyLAB's willingness to allow a data science team to investigate the applications of various new methods and technologies in the field of text-mining is very much appreciated.

References

1. Carranza Tejada, G.N.: A study of word embeddings and support vector machines for emotions and sentiments recognition. Tech. rep. (2020)
2. Ettinger, A.: What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. Transactions of the Association for Computational Linguistics **8**, 34–48 (2020)
3. Goldberg, Y.: Assessing bert's syntactic abilities. arXiv preprint arXiv:1901.05287 (2019)
4. Kassner, N., Kroje, B., Schütze, H.: Pre-trained language models as symbolic reasoners over knowledge? arXiv preprint arXiv:2006.10413 (2020)
5. Kassner, N., Schütze, H.: Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. arXiv preprint arXiv:1911.03343 (2019)
6. Khandelwal, A., Sawant, S.: Negbert: A transfer learning approach for negation detection and scope resolution. arXiv preprint arXiv:1911.04211 (2019)
7. Niven, T., Kao, H.Y.: Probing neural network comprehension of natural language arguments. arXiv preprint arXiv:1907.07355 (2019)
8. Novielli, N., Girardi, D., Lanubile, F.: A benchmark study on sentiment analysis for software engineering research. In: 2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR). pp. 364–375. IEEE (2018)
9. Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.H., Riedel, S.: Language models as knowledge bases? arXiv preprint arXiv:1909.01066 (2019)
10. Ribeiro, M., Singh, S., Guestrin, C.: " why should i trust you?" explaining the pre-dictions of any classifier. Proceedings of the 22ndACM SIGKDD international conference on knowledge discovery and data mining p. 1135–1144 (August 2016)
11. Rosenthal, S., Farra, N., Nakov, P.: SemEval-2017 task 4: Sentiment analysis in Twitter. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 502–518. Association for Computational Linguistics, Vancouver, Canada (Aug 2017). https://doi.org/10.18653/v1/S17-2088, `https://www.aclweb.org/anthology/S17-2088`
12. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 1631–1642 (2013)
13. Tang, D., Wei, F., Qin, B., Yang, N., Liu, T., Zhou, M.: Sentiment embeddings with applications to sentiment analysis. IEEE transactions on knowledge and data Engineering **28**(2), 496–509 (2015)
14. Yadollahi, A., Shahraki, A.G., Zaiane, O.R.: Current state of text sentiment analysis from opinion to emotion mining. ACM Computing Surveys (CSUR) **50**(2), 1–33 (2017)
15. Zhao, Y., Bethard, S.: How does bert's attention change when you fine-tune? an analysis methodology and a case study in negation scope. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4729–4747 (2020)

MoveRL: To A Safer Robotic Reinforcement Learning Environment

Gaoyuan Liu, Joris De Winter, Bram Vanderborght, Ann Nowé, and Denis Steckelmacher

Vrije Universiteit Brussel
gaoyuan.liu@vub.be

Abstract. The deployment of Reinforcement Learning (RL) on physical robots still stumbles on several challenges, such as sample-efficiency, safety, reproducibility, cost, and software platforms. In this paper, we introduce MoveRL, an environment that exposes a standard OpenAI Gym interface, and allows any off-the-shelf RL agent to control a robot built on ROS, the Robot OS. ROS is the standard abstraction layer used by roboticists, and allows to observe and control both simulated and physical robots. By providing a bridge between the Gym and ROS, our environment allows an easy evaluation of RL algorithms in highly-accurate simulators, or real-world robots, without any change of software. In addition to a Gym-ROS bridge, our environment also leverages MoveIt, a state-of-the-art collision-aware robot motion planner, to prevent the RL agent from executing actions that would lead to a collision. Our experimental results show that a standard PPO agent is able to control a simulated commercial robot arm in an environment with moving obstacles, while almost perfectly avoiding collisions even in the early stages of learning. We also show that the use of MoveIt slightly increases the sample-efficiency of the RL agent. Combined, these results show that RL on robots is possible in a safe way, and that it is possible to leverage state-of-the-art robotic techniques to improve how an RL agent learns. We hope that our environment will allow more (future) RL algorithms to be evaluated on commercial robotic tasks.
Github repository: https://github.com/Gaoyuan-Liu/MoveRL

1 Introduction

Reinforcement Learning is a Machine Learning approach that allows an agent to learn what action to execute in which situation, to maximize a scalar reward [14]. On robots, Reinforcement Learning has the potential of allowing to learn near-optimal controllers on challenging tasks, on which classical methods such as planning are not applicable, for instance due to the unavailability of a good model, or high stochasticity or unexpected events around the robot. However, in practice, Reinforcement Learning is not often used on robots.

Several challenges currently prevent the use of Reinforcement Learning on robots, such as safety, sample-efficiency, the ease of implementation of RL on

robots from a software perspective, and the trust designers must have in RL to use it. In this paper, we propose a new OpenAI Gym [3] environment that allows real-world robotic experiments to be performed, addressing these two challenges:

Software compatibility with robots. Existing Reinforcement Learning environments that have robots in mind, such as the Gym Mujoco environments [3], the DeepMind control suite [26], or PyBullet environments [29], implement environment-specific robotic arms or bodies (not industry-standard robots), using embedded simulators (not a connection to an industry-standard simulator). As such, these environments can be used to show that RL works on robots in theory, but do not help implementing RL on a real-world robot. Our main contribution, MoveRL, interfaces an RL agent with the Gym API to ROS, the Robot OS, used by industry-standard simulators (such as Gazebo) and robots. This allows direct learning on the robot, or easy transfer of an agent learned in simulation to a physical robot (without having to re-implement anything).

Safety The Robot OS comes with many packages that allow to build complete robotic systems, with planning, collision avoidance, simultaneous localization and mapping, ... In this paper, we use MoveIt [4] to transform an action selected by an RL agent into a motion plan for a robot, while avoiding collisions with obstacles. MoveIt gets its knowledge about obstacles from the ROS network, which means that it is inherently compatible with simulators (that know where obstacles are) and depth cameras, that produce the same information on real robots [11].

Our empirical results in the Gazebo simulator, using a simulated real-world robot (the Franka Emika Panda manipulator), show that combining an unmodified implementation of PPO [24] from the stable-baselines3 [22] with a ROS environment is possible, and that leveraging MoveIt for action execution allows to prevent almost every collision, even in the early stages of learning.

2 Notations

The Reinforcement Learning literature considers an agent that executes actions in a Markov Decision Process, defined by a tuple $\langle S, A, R, T, \mu_0, \gamma \rangle$. S is the state space, that can be either discrete or continuous. In this paper, we consider continuous state-spaces, in which each state is a vector of several real values. A is the action space. In this paper, we consider a continuous action space, in which each action is a vector of real values. $R(s, a, s')$ is the reward function, that produces a single real value after a transition from state s to s', caused by the execution of action a. $T(s, a)$ is the transition function, that maps a state and an action to a new state. μ_0 is the initial state distribution, that defines in which state the agent may start an episode, and $\gamma < 1$ is a real value, the discount factor.

Most Reinforcement Learning literature follows the notation described above. However, roboticians use other notations, that appear in the literature related to

ROS and MoveIt, and that we sometimes use in this paper when interfacing with these components. We provide a brief summary of the differences of notation in the table below:

RL	Meaning	Motion Planning
s	State (observation)	q (if joint angles) p (if end-effector position)
a	Action	q_i (target joint angles)
r	Reward	r or c (cost $c = -r$)

3 Related Work

Our main contribution allows a Reinforcement Learning agent to interface with the Robot OS, for easy control of simulated or physical robots, with the use of a motion planner to ensure safety. We now provide a related work review of other approaches at robotic environments for Reinforcement Learning, techniques that allow to make a Reinforcement Learning agent safer, and motion planning libraries.

3.1 RL Robotic Environment

To tackle various challenges in robot RL [15], numerous robotic RL environments are developed with different platforms. A brief survey of robotic RL environment frameworks can be found in [13] and [9]. Each work emphasises specific merits with regards to particular issues. In this paper, we only review frameworks which are wildly accepted as benchmark, and particularly, we discuss how they consider safety when learning.

Mujoco To improve the reproducibility in RL robotics research, SURREAL [8] is built on the MuJoCo simulation environment and physics engine [27]. Mujoco is widely used for RL environments, and frameworks with the same physics engine can be found in [21, 31, 1]. Such projects usually focus on theoretical RL research, and lack compatibility with robotic software such as the Robot OS. Moreover, the use of Mujoco requires a license id (free for academic purposes, paid otherwise).

PyBullet PyBullet is an open-source physics engine, used by [30] to implement several Reinforcement Learning environments in simulated 3D spaces. These environments allow the agent to control every joint of the robots, but do not provide any safety mechanism or collision avoidance. An RL environment for a quadcopter is developed with PyBullet by [19], but collision avoidance is not considered, even though it appears crucial for a quadcopter.

Gazebo Gazebo is an open-source simulator with a graphical interface, and an interface to the Robot OS ROS [6]. A Gym environment for interacting with Gazebo is proposed in [16], but collisions are allowed to happen in the simulator, which makes replacing the simulator with a real-world robot impractical. However, because Gazebo and ROS have large communities that developed many industry-proven tools, we base our main contribution on these two pieces of software, and add MoveIt for collision avoidance.

3.2 Safe Reinforcement Learning

RL safety is normally defined as a mechanism which can ensure reasonable system performance and/or respect safety constraints during the training or validation processes.

Definition and Survey RL safety approaches can be categorized in two classes: tuning the optimization criterion of the algorithm to encourage safe behavior, and directly intervening on the exploration of the agent to prevent unsafe actions from being executed.

With the optimization approach, maximizing the long-term reward can generate statistically safer policy, but does not necessarily avoid the rare occurrences of damage, neither ensures safety during training. The exploration approach provides a *shielding* mechanism that modifies or prevents unsafe actions [10]. Several approaches to Safe RL, belonging to the two classes described above, are reviewed in [28].

Safe exploration In this paper, we focus our attention onto Safe RL approaches that consider physical issues, and in particular prevent physical damage. In a danger-sensitive learning environment, such as robotics, the importance of damage avoidance is higher than obtaining high rewards. Therefore, a shielding layer maintaining zero-constraint-violations throughout whole learning process is necessary.

[23] introduce *safe exploration*, more specifically Constrained Reinforcement Learning, and address two challenges: 1) the difficulty of designing reward functions that nicely balance punishing unsafe actions, and encouraging the agent to learn the desired skill; 2) the fact that eventually learning the optimal safe policy does not guarantee that no unsafe action has been performed while learning.

[5] consider that some states can be identified as unsafe, and propose a method to avoid these states. In [20], a safety layer is applied in a real-robot system, the safety layer modifies possibly risky actions to the closest valid alternatives which satisfy safety constrains, but such constraints are difficult to define especially when the environment is noisy or uncertain. Similar structured safety guarantee is also utilized in [2].

When positioning this paper in relation to existing work, it is important to note that existing work focuses on preventing the execution of specific unsafe actions, and let the designer define what an unsafe action is. In this paper, we

use MoveIt to automatically detect what would be unsafe actions, freeing the designer from this task. Moreover, existing work considers that moving from a safe state to a safe state is safe. This is not the case in practice, as we explain in Section 4.7: the path between two safe states may go through a wall, and therefore be unsafe. Our contribution detects these unsafe actions.

3.3 Path Planning

Path planning is one of the most fundamental problems in autonomous robotics, particularly, in the scenarios where robots have to execute tasks in an environment with obstacles. Sampling-based methods offer a solution to overcome the complexity of deterministic robot planning algorithms for a robots with many degrees of freedom (many joints). A comprehensive survey can be found in [7]. An open-source library for sampling-based motion planning OMPL (Open Motion Planning Library) is proposed by [25], and is integrated in an open-source framework, MoveIt!, that offers an state of the art path planning based on several well-known libraries. MoveIt also integrates implementations of useful robotics functions, such 3D perception, kinematics calculation and control [1].

4 Contribution

Our main contribution is a Gym environment, that allows to interface unmodified Reinforcement Learning agents written in Python with a simulated or physical robot exposed on ROS-Noetic, the Robot OS, a collection of libraries and network protocols that allow components of robotic systems (hardware, software, planners, ...) to communicate in an industry-standard way. We also leverage MoveIt, a state-of-the-art motion planner, for efficient action execution and collision avoidance.

The general architecture of our main contribution, MoveRL, is depicted in Figure 1. Every time-step, the agent receives an observation and reward from the environment, and selects a *raw action*, a desired position of the robot, not yet guaranteed safe. The raw action passes through a collision monitor, based on MoveIt, that observes the current position of obstacles and verifies the action. Verifying the action relies on the possibility to simulate its outcome, which is possible on physical robots by using *co-simulation* (a simulated version of the robot runs in parallel with the physical robot, an approach very common in industrial robotics and transparently supported by ROS). In this paper, we only use pure simulation, and leave co-simulation with a physical robot to future work.

If the raw action will cause a collision, it is discarded, leading to no movement of the robot for this time-step, and a punishment given to the Reinforcement Learning agent. If the raw action is safe, the planner computes a collision-free path to guarantee the safety during execution. Our action shielding mechanism ensures that no collision happens when the agent moves.

[1] https://MoveIt.ros.org/

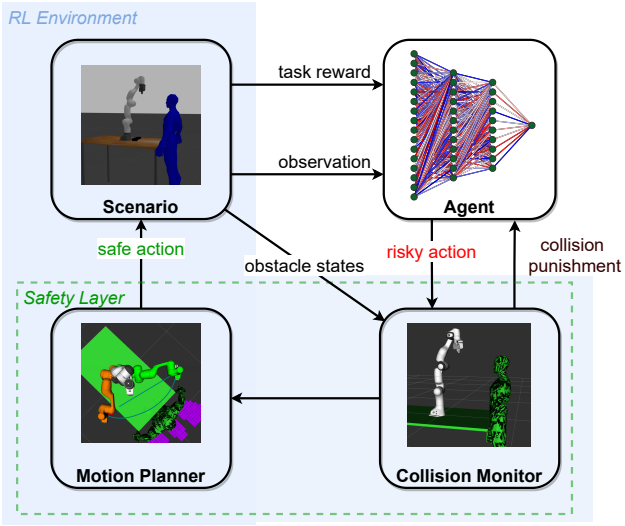


Fig. 1. Our MoveRL framework. Our safety layer leveraging the MoveIt motion planner contains 2 modules: 1) a collision monitor, that detects actions that lead to collisions, and 2) a motion planner, that plans a collision-free path to reach the target locations encoded in (previously-identified) safe actions.

We now detail all the software components of our proposed MoveRL. Our implementation is available at https://github.com/Gaoyuan-Liu/MoveRL.

4.1 The Gym environment

To be compatible with standard RL algorithm implementations, such as in Stable Baseline 3 [22], our contribution needs to be implemented as an OpenAI Gym [3] environment. A Gym environment is a Python class that contains attributes that describe its state and action spaces (both `Box` in our case), and methods that allow actions to be executed in the environment. The *reset* method resets the environment to an initial state, and returns the first observation of the episode. The *step* method takes an *action* as input, passes to ROS and MoveIt for safe execution, and produces a new state (*observation*) and *reward*. We describe all these steps in more detail later. A *done* signal is also returned by *step*, and allows the environment to choose when an episode should terminate.

4.2 Observation Space

Since we consider kinematics observation, we make an assumption that the position of obstacles can be detected by sensors, or is available in simulation. The observation space contains two parts: the state of the robot, and the state of the obstacles. For the robot state, we developed two environments with two different

kinds of observation: joint angles $\mathbf{q} = [q_1, q_2, ..., q_n]$ for an n degrees-of-freedom robot, and end-effector position and orientation $\mathbf{p} = [p_x^{ee}, p_y^{ee}, p_z^{ee}, o_x^{ee}, o_y^{ee}, o_z^{ee}, o_w^{ee}]$, for tasks in which the robot has an end-effector such as a gripper. For obstacle state, the agent can observe the position and orientation of the obstacles: $[p_x^{obs}, p_y^{obs}, p_z^{obs}, o_x^{obs}, o_y^{obs}, o_z^{obs}, o_w^{obs}]$. Our environment class can adjust the size of state space according to the number of obstacles in the simulation.

Note that the agent observes the position and orientation of the obstacles (cylinders, spheres, rods, cubes), but not their shape or dimensions. This is not a problem, as an RL agent is perfectly able to learn what positions in relation to the center and orientation of an obstacle will translate to negative rewards. So, the agent sorts of learns the shape of the obstacles by feel, and does not need to be provided that information.

4.3 Action Space

Our environment exposes a continuous action space, for which actions are vectors of real values. More precisely, we consider that the action produced by the agent is a target configuration of the robot, so a list of real values that define the angle at which every joint of the robot must be set. Robotics libraries call this set of angles q_i, and the Reinforcement Learning literature calls this a. The action space is constrained by the physical abilities of the robot, with joint position limits $q_{i,\text{limit}}$ and joint velocity limits $\dot{q}_{i,\text{limit}}$.

The physical constraint on the speed of a joint requires careful engineering of how the agent produces an action. Given a time-step duration Δt, we must ensure that the action $q_{i,\text{cmd}}$ produced by the agent, and sent to the environment, differs (in absolute value) from the previous action by at most $\Delta t \cdot \dot{q}_{i,\text{limit}}$, for every element of $q_{i,\text{cmd}}$. We must also ensure that the action $q_{i,\text{cmd}}$ is part of the allowed range of joint angles $[q_{i,\text{min}}, q_{i,\text{max}}]$.

We implement these constraints as follows: the policy of the agent produces the change in joint positions $\Delta q_{i,\text{cmd}}$, instead of the absolute value of the joint positions $q_{i,\text{cmd}}$. Then, we clip $\Delta q_{i,\text{cmd}}$ to $[-\Delta t\dot{q}_{i,\text{limit}}, \Delta t\dot{q}_{i,\text{limit}}]$, produce $q_{i,\text{cmd}} = q_{i,\text{prev timestep}} + \Delta q_{i,\text{cmd}}$, and clip $q_{i,\text{cmd}}$ to the range $[q_{i,\text{min}}, q_{i,\text{max}}]$. This clipped value is sent to the environment, that uses MoveIt to detect and avoid collisions.

4.4 Why do we need sequences of actions?

Most tasks on which we evaluate our framework consist of moving the end effector of a robot to a specific target location. Given the action set described above, it may seem logical that only one action is necessary for that: putting the robot in the pose that puts the end effector at the target location. However, in practice, a sequence of actions is needed for the following reasons:

– The time-step has a fixed duration and the robot cannot move infinitely quickly, so the actions have to progressively bring the robot close to the target location;

– Even if state of the art, MoveIt has difficulties planning motions on long distances, especially when there are concave obstacles in the scene. Reinforcement Learning is particularly useful in this case, as its optimization of the discounted sum of rewards allows the agent to take actions that move away from the goal in the short term, but allow to reach it in the long term.

4.5 Reward Function

The reward function is customized for each specific task, but always consists of the sum of two terms: a task-specific reward and a task-agnostic safety term, $r = r_{\text{task}} + r_{\text{safety}}$.

We describe r_{task} in the next sections. r_{safety} is 0 when an action would cause no collision, and some negative constant when an action is detected as being unsafe (and cancelled). The choice of the constant is described in our experiments, and needs to be large enough that the agent learns to avoid states that can potentially lead to collisions (especially when there are moving obstacles), but not too much, so that the agent does not become too conservative (and learns a policy that remains immobile, for instance).

4.6 Initial States and Termination

Every episode, we initialize the simulated robot to a random pose, to ensure good exploration. The episode terminates when the end effector of the robot reaches a pre-defined goal position, or after 100 time-steps.

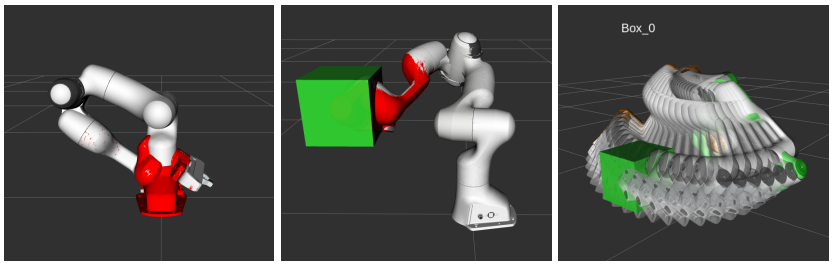
4.7 Safety Guarantee

After having described the different components of our environment (state space, action space, ...), we now discuss the different types of collisions that can be detected by MoveIt, and provide details on how we query MoveIt from a Gym environment. A full description with code would go beyond the page limit of this paper, but the complete source code that we use in our experiments is available on Github (link in the abstract).

Collision Types To comprehensively consider the potential risk during training, we categorize collisions into three types:

(a) self collision: two parts of robot itself collide with each other; (b) pose collision: the commanded configuration $q_{i,\text{cmd}}$ collides with objects in the environment; (c) path collision: the direct path between two configurations contains collisions with objects in the environment. Figure 2 shows examples of these three types of collisions.

Avoiding self-collision and pose collision during learning can be achieved by constrained inverse kinematics and state-validation checking at each time-step. However, the avoiding path collisions is more challenging, and tends to be neglected in the Safe RL literature since it is difficult to do state validation



(a) (b) (c)

Fig. 2. (a) Self collision (b) Pose collision (c) Path collision

checking continuously. Therefore, even when the state for two adjacent steps are safe, the direct path (without planning) can still collide with obstacles. We give an unified solution to avoid the aforementioned three types of collision, which is integrating MoveIt as an safety layer in the RL environment.

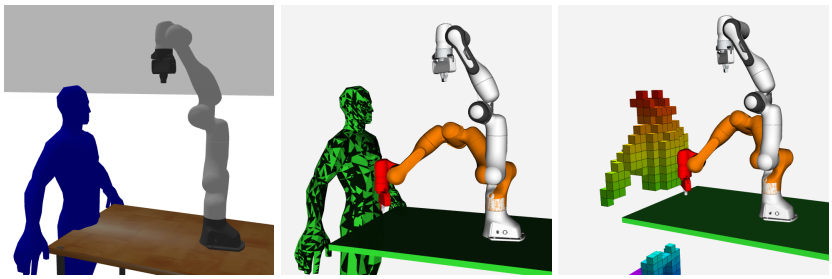
Collision Detection The MoveIt provide package `Planning Scene` allows to manage an abstract representation of the environment surrounding a simulated or physical robot. This environment can contain obstacles of two possible types: scene object and octomap [12].

Scene objects have an explicit shape, such as a 3D mesh or a primitive shape (cylinder, box, sphere). It is used when a coarse obstacle is enough, for instance a big cylinder around a human, to encoder a general area that has to be avoided. An Octomap is built from a depth camera (or produced by a simulator), and allows to precisely measure the presence of an obstacle around the robot, without having to model it. The trade-off between precision and efficiency can be adjusted with the resolution parameter of the octomap.

Once the Planning Scene has been defined (and kept updated by the simulator or sensors, using ROS network messages that the Gym environment does not even need to bother with), MoveIt is able to detect collisions using libraries such as the FCL (Flexible Collision Library) [18].

We stress that the use of ROS allows to transparently interface our Gym environment with many well-regarded robotic packages, Planning Scene being only one. Other packages allow to stream updates to the position of the obstacles from a variety of sensors (and are usually shipped with the sensors), or to visualize various aspects of the scene (for instance, visualizing how a physical robot senses its surrounding). Figure 3 shows that it is possible to visualize a textured 3D render of a scene, along with information about the obstacles in it, and what motion planning has to be performed.

Path Planner It's worth noting that the direct path between two valid poses can still contain collisions, which we term as path collision. To avoid path collisions, a local planner is necessary, and will run every time-step, to produce a



(a) (b) (c)

Fig. 3. The collision detection methods in MoveIt. (a) Gazebo simulation (b) Collision detection with scene object (c) Collision detection with octomap. The white arm indicates current pose, the orange arm indicate the commanded pose, and the red parts indicate the collision links.

full motion from the pose of the robot on one time-step, to its pose at the next time-step.

For each time-step, given a valid action command, a planner plans a collision-free local path based on the present knowledge of the position of obstacles. In order to reduce the time consumption of training and minimize the planning delay, our primary consideration for choosing the most appropriate planner is efficiency and completeness. Therefore, we evaluated planners available in MoveIt by their solving time and path length, which can reflect the planners' efficiency in either planning and execution phases [17].

The planners' time consumption and planned path length on a benchmark task (putting a robotic arm into a desired pose) are shown in Figure 4. We note that our objective is not to identify the absolute best planner, but to provide an informed choice of planner for our Reinforcement Learning experiments. RRT and its derivative show merits in both planning time and path length. Therefore, we choose the RRT planner in our experiments.

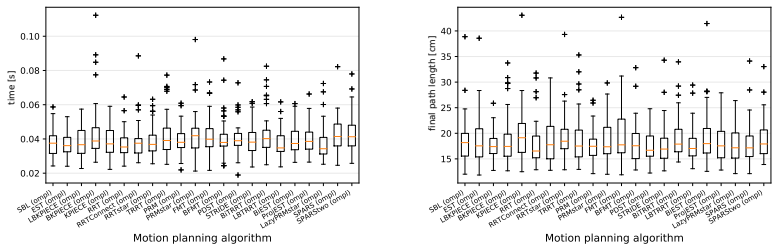


Fig. 4. *Left:* Planner planning-time comparison, *right:* Planner final path length comparison.

Goal-Reaching Task We now define a goal-reaching task, and detail how it is implemented with MoveIt. To achieve the goal-reaching task, we define a dense reward (non-zero whenever there is movement during a time-step), that is proportional to the change in distance between the end-effector's current position and the goal position. The task reward can be formalized as:

$$r_{\text{task}}(s) = \begin{cases} \kappa \Delta d(s) + r_{\text{goal}} & \text{if } s \text{ is close to } s_{\text{goal}} \\ \kappa \Delta d(s) & \text{otherwise} \end{cases}$$

where $r_{\text{task}}(s)$ is the task reward given to the agent when reaching state s, $\Delta d(s)$ is the distance between the end-effector in state s and the target end-effector location, r_{goal} is a fixed positive reward given when the target location is reached, and κ is a weighting constant, allowing to balance r_{task} and r_{safety}. The actual values of r_{goal} and κ are given in the next section.

5 Experiment

While our main contribution is a Gym environment that allows to learn tasks in ROS-based robotic environments with standard Reinforcement Learning algorithms, we also provide experimental results, that show that:

- Our framework, that we call MoveRL, works and actually allows an unmodified PPO agent to learn a task;
- Collisions can indeed be avoided, thanks to MoveIt, which allows simulated robots to be replaced with physical robots if need be.

5.1 Learning Scenarios

In our experiments, the robot learns to fetch the goal point with its end-effector by adjusting 7 joint angles. We consider 3 scenarios around this task, that differ in what kinds of obstacles are around the robot:

1. **Table**: The table holding the robot is the only exterior obstacle, thus self-collisions are considered as the major risks in this scenario.
2. **Human**: The robot and a human worker share a single workspace. The goal point is located between the human and the robot. Self-collisions and pose collisions would be the major risks. The human is modelled with basic shapes such as cylinders.
3. **Case**: The robot has to reach a goal location inside a box/case (walls with an opening on top). Finding how to enter the case is challenging in this task and benefits from the use of Reinforcement Learning. The thin walls of the case lead to possible path collisions (in addition to self-collisions and pose collisions).

The 3 training scenarios are illustrated in figure 5. Our Github repository contains Gazebo world files for all 3 scenarios.

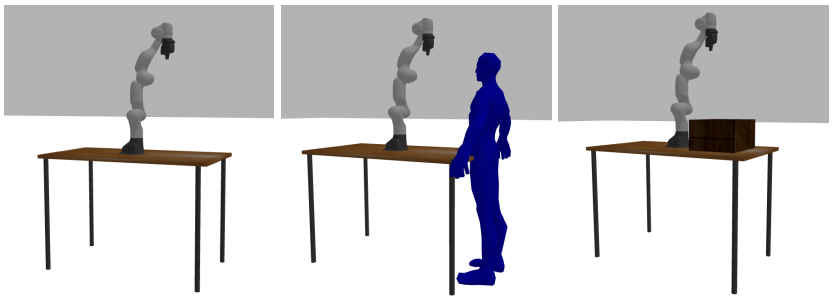


Fig. 5. From left to right: table world, human world and case world, and the first row shows the simulation environment in gazebo while the second row is the state presentation (robot and obstacles) in rviz.

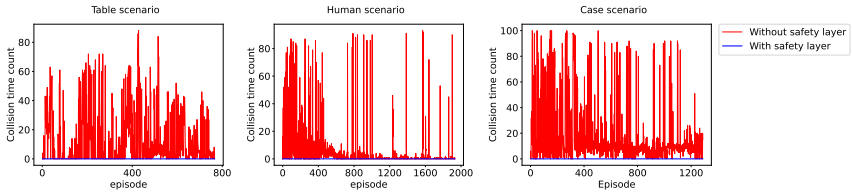


Fig. 6. Number of collisions occurring per episode, with and without collision avoidance with MoveIt. The blue line indicates that our safety layer, based on MoveIt, successfully prevents collisions throughout the learning process.

5.2 Learning Algorithm

We use a Reinforcement Learning agent from the Stable-Baselines 3 [22], that contains a set of reliable RL algorithms including A2C, DDPG, PPO, SAC and TD3. We choose PPO, as it is highly popular, compatible with continuous actions, and has many implementations. In this paper, we focus on showing that RL with ROS is possible, we do not aim at evaluating which RL algorithm performs the best. The hyper-parameters that we use for PPO in our experiments are the default values used Stable-Baselines 3, as of August 23rd, 2021, with the following changes: the policy network is MlpPolicy, the learning rate is 0.0005, the batch size is 200, and the number of steps between policy updates is 100.

5.3 Results

To evaluate our safety layer, we compare how a PPO agent learns with and without our collision avoidance method. We observe that our safety layer successfully prevents collisions, and has no negative impact on sample-efficiency:

Figure 6 shows that enabling our safety layer successfully prevents collisions, and that disabling our safety layer leads to a large amount of collisions.

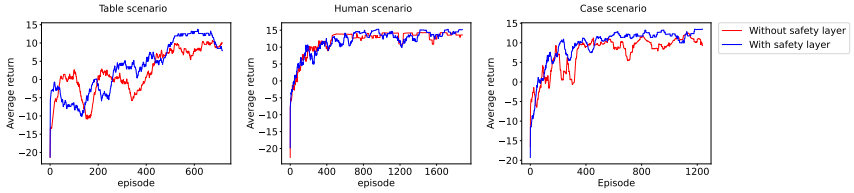


Fig. 7. Learning curves in each learning scenario. We confirm that our safety layer, that successfully prevents collisions, has no negative impact on sample-efficiency (the red and blue curves have the same shape). This shows that safety does not come at the expense of sample-efficiency with our MoveRL framework.

Figure 7 presents the learning curves in our three scenarios, with and without our collision avoidance method. Avoiding collisions does not appear to have any negative impact on the agent, as the learning curves are comparable. If it has any effect, it would be a slight increase in sample-efficiency, as seen in the Case scenario. We are happy with this result, as it shows that safety in Reinforcement Learning does not come (in our case) at the cost of sample-efficiency and final policy quality.

6 Conclusion

In this paper, we presented MoveRL, a Reinforcement Learning Gym environment for robotic manipulators, that builds the widely-used ROS platform for simulated and physical robots. Thanks to the dynamism of the ROS community, advanced algorithms for planning, obstacle detection and collision avoidance are available. We leverage them in our environment to produce a method for safe Reinforcement Learning on robots. Our experiments show that our safety mechanism indeed prevents collisions while an un-modified PPO agent learns a simulated robotic task, and that our method has no negative impact on sample-efficiency.

While the deployment of our method on a physical robot remains as future work, we hope that our new software and method will allow Reinforcement Learning researchers to more easily evaluate their methods on simulated real-world robots (as opposed to unrealistic robots as available in the Gym Mujoco tasks, for instance), and will allow robotic engineers to evaluate Reinforcement Learning for the tasks in which classical planning methods show limitations.

Acknowledgments

The first author is supported by the China Scholarship Council (CSC). The second author is supported by the Flemish Government under the Flemish AI Program (*Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen*).

Bibliography

[1] Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. Robel: Robotics benchmarks for learning with low-cost robots. In *Conference on Robot Learning*, pages 1300–1313. PMLR, 2020.

[2] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[4] Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit![ros topics]. *IEEE Robotics & Automation Magazine*, 19(1):18–19, 2012.

[5] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

[6] Brian Delhaisse, Leonel Rozo, and Darwin G Caldwell. Pyrobolearn: A python framework for robot learning practitioners. In *Conference on Robot Learning*, pages 1348–1358. PMLR, 2020.

[7] Mohamed Elbanhawi and Milan Simic. Sampling-based robot motion planning: A review. *Ieee access*, 2:56–77, 2014.

[8] Linxi Fan, Yuke Zhu, Jiren Zhu, Zihua Liu, Orien Zeng, Anchit Gupta, Joan Creus-Costa, Silvio Savarese, and Li Fei-Fei. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In *Conference on Robot Learning*, pages 767–782. PMLR, 2018.

[9] Diego Ferigo, Silvio Traversaro, Giorgio Metta, and Daniele Pucci. Gym-ignition: Reproducible robotic simulations for reinforcement learning. In *2020 IEEE/SICE International Symposium on System Integration (SII)*, pages 885–890. IEEE, 2020.

[10] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[11] Stefan Grushko, Aleš Vysockỳ, Vyomkesh Kumar Jha, Robert Pastor, Erik Prada, L'ubica Miková, Zdenko Bobovskỳ, Jiří Suder, Zdeněk Zeman, Jakub Milotek, et al. Tuning perception and motion planning parameters for moveit! framework. 2020.

[12] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206, 2013.

[13] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[14] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[15] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[16] Nestor Gonzalez Lopez, Yue Leire Erro Nuin, Elias Barba Moral, Lander Usategui San Juan, Alejandro Solano Rueda, Víctor Mayoral Vilches, and Risto Kojcev. gym-gazebo2, a toolkit for reinforcement learning using ros 2 and gazebo. *arXiv preprint arXiv:1903.06278*, 2019.

[17] Mark Moll, Ioan A Sucan, and Lydia E Kavraki. Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization. *IEEE Robotics & Automation Magazine*, 22(3):96–102, 2015.

[18] Jia Pan, Sachin Chitta, and Dinesh Manocha. Fcl: A general purpose library for collision and proximity queries. In *2012 IEEE International Conference on Robotics and Automation*, pages 3859–3866. IEEE, 2012.

[19] Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P Schoellig. Learning to fly–a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. *arXiv preprint arXiv:2103.02142*, 2021.

[20] Martin Pecka and Tomas Svoboda. Safe exploration techniques for reinforcement learning–an overview. In *International Workshop on Modelling and Simulation for Autonomous Systems*, pages 357–375. Springer, 2014.

[21] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

[22] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3. https://github.com/DLR-RM/stable-baselines3, 2019.

[23] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7, 2019.

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[25] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[26] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[27] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[28] Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained markov decision processes. In *International Conference on Machine Learning*, pages 9797–9806. PMLR, 2020.

[29] Xintong Yang, Ze Ji, Jing Wu, and Yu-Kun Lai. An open-source multi-goal reinforcement learning environment for robotic manipulation with pybullet. *arXiv preprint arXiv:2105.05985*, 2021.

[30] Xintong Yang, Ze Ji, Jing Wu, and Yu-Kun Lai. An open-source multi-goal reinforcement learning environment for robotic manipulation with pybullet. *arXiv preprint arXiv:2105.05985*, 2021.

[31] Yuke Zhu, Josiah Wong, Ajay Mandlekar, and Roberto Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

Proximal Policy Optimisation for a Private Equity Recommitment System

Emmanuel Kieffer[1], Frédéric Pinel[1], Thomas Meyer[3], Georges Gloukoviezoff[2], Hakan Lucius[2], and Pascal Bouvry[1]

[1] University of Luxembourg, Esch-sur-Alzette, Luxembourg
firstname.lastname@uni.lu
[2] European Investment Bank, Luxembourg, Luxembourg
{g.gloukoviezoff,h.lucius}@eib.org
[3] SimCorp Luxembourg SA, Luxembourg, Luxembourg
thomas.meyer@simcorp.com

Abstract. Recommitments are essential for limited partner investors to maintain a target exposure to private equity. However, recommitting to new funds is irrevocable and expose investors to cashflow uncertainty and illiquidity. Maintaining a specific target allocation is therefore a tedious and critical task. Unfortunately, recommitment strategies are still manually designed and few works in the literature have endeavored to develop a recommitment system balancing opportunity cost and risk of default. Due to its strong similarities to a control system, we propose to "learn how to recommit" with Reinforcement Learning (RL) and, more specifically, using Proximal Policy Optimisation (PPO). To the best of our knowledge, this is the first attempt a RL algorithm is applied to private equity with the aim to solve the recommitment problematic. After training the RL model on simulated portfolios, the resulting recommitment policy is compared to state-of-the-art strategies. Numerical results suggest that the trained policy can achieve high target allocation while bounding the risk of being overinvested.

Keywords: Reinforcement learning · Private Equity · Control system.

1 Introduction

Private equity is an alternative asset class which refers to direct investments in non-listed companies made at different stages of their development to create added value. These companies are then sold few years later with the expectation to obtain a significant capital gain. Early investments in strong performing companies help them to develop their business and make them more profitable. Contrary to the public equity market, private equity investments are not easily accessed as stocks and bonds. Recently, private equity has been included in the portfolios of institutional investors such as pension funds, sovereign wealth funds, etc. These institutional investors have been building sizable allocation by investing "indirectly" to private companies through private equity funds. Indeed,

managing such a less traditional asset class requires a high level of expertise to properly enter and exit direct investments. This explains their preferred modus operandi to invest indirectly as so-called limited partners (LP) through limited partnership funds in which they commit a certain amount of capital for a given period of time. Commitments are irrevocable and called at the discretion of the fund's management, i.e., the general partner (GP), to decide how investments should be realised. The committed capital is gradually draw down during the so-called investment period which last several years. To complicate matters, stakes in these funds are illiquid [7] which enforce LP investors to be extremely cautious when it comes to recommit into new funds to limit the risk of default. Generally, the committed capital is an upper-bound of the total capital finally called by a fund. A significant part ($\approx 10\%$) of the initial capital is generally never invested as described in [18]. Furthermore, committed capital waiting to be called is generally pictured as dry powder. Prequin [4] reported in November 2020 that North American private equity firms are sitting on almost $980bn in reserves. This uncalled capital dramatically impacts investors' exposure (see [12]). In practice, LP investors therefore run so-called overcommitment strategies, i.e., committing more capital in aggregate than actually available as dedicated resources, with the gap expected to be filled by future distributions from investments made in other existing funds. These strategies thus increase the liquidity risk when the fund is only few years old when the likelihood to be called is the highest. LP investors need to setup a commitment-pacing strategy, i.e., on how to size and time their commitments, in order to achieve and maintain a target allocation while complying with the liquidity constraints imposed by the uncalled capital. As reported in [3] and [9], few investigations have been engaged to evaluate the cost of maintaining uncalled capital. This is the reason why the current existing models still remain rudimentary and depend on spreadsheet-based and "trial-and-error" approaches. These manually-designed strategies are often error-prone and naive although the opportunity cost, i.e., the cost of being underinvested, and the risk of default in case of overinvestment can be very damaging for LP investors.

In this work, we propose to investigate an approach relying on Reinforcement Learning to learn how to size and time dynamic recommitments. The latter can be formulated as a RL problem to discover reliable recommitment policies using a Proximal Policy Optimisation algorithm. Recommitment policies can be assimilated as control policies which should maintain a target allocation minimizing the opportunity cost while preserving investors from the risk of default.

The remainder of this paper is organized as follows. The next section provides a state of the art on existing recommitment strategies. Section 3 introduces formally the Private Equity Recommitment Problem (PERP). Section 4 described the Proximal Policy Optimisation algorithm applied on the RL version of the PERP introduced in Section 5. Experiment setups and results are discussed in

[4] https://www.preqin.com/insights/research/blogs/what-private-equitys-record-dry-powder-haul-means-for-the-industry

Section 6 and 7. Finally, the last section provides our conclusions and proposes some possible perspectives.

2 Related works

Recommitment strategies are essential to keep investors constantly invested at some target allocation. To the best of our knowledge, few studies have tried to model this as an optimisation problem. They generally rely on some rules of thumb lacking robustness and flexibility. In [4], authors considered that the entire private equity allocation should be recommitted to new funds every year without taking into account past portfolios evolution. Nevin et al. in [11] based their recommitment strategy on average rates of distributions and commitments. New commitments should be made if the committed capital does not reach a target threshold to compensate the difference. This strategy assumes constants rates which seems very illusory over time. In [18], de Zwart et al. proposed recommitment strategies for funds aiming to maintain stable the exposure to PE. The strategy's key feature is the level of new commitments in a given period which depends on the current portfolio's characteristics. Importantly, de Zwart's strategies does not require to forecast funds'cashflows. Although they consider 100% PE portfolios, their last suggested strategy is a first attempt to design dynamic recommitment strategies relying on past portfolio development. Finally, Oberli et al. in [12] extended de Zwart's work to multi-asset class portfolios including stocks and bonds. These two last contributions solely rely on handcrafted recommitment strategies to control the investment degree (ID), i.e., PE exposure. While they are innovative and improving attempts without the need to forecast future cashflows, they have been built on specific and limited datasets with given market conditions. Building recommitment strategies in various market conditions is a challenging task. In this work, we investigate Reinforcement Learning to discover promising recommitment policies using the policy-based PPO algorithm. Policy-based algorithms [13, 15] have been motivated by the fact that solving a RL problem is all about finding a sequences of actions even for value-based algorithms [10, 6]. Discovering and predicting the best actions avoid the computational burden to compute all state values. Besides, when the action space is continuous or very large, policy-based approaches are more attractive than values as we do not need to solve an optimisation problem to select the best action.

3 Problem description

This section describes the Private Equity Recommitment (PERP) by considering a single LP investor owning a 100% private equity portfolio. To minimize the opportunity cost, the investor's primary target is to remain fully invested while avoiding cash shortage. Let us define $\mathcal{P}(t) = \{f\}_{i=1}^{M}$ the set of active funds in the portfolio at time t. In order to measure its degree of investment, the fraction of total allocated capital that is actually invested can be computed as follows:

$$ID(\mathcal{P},t) = \frac{\sum\limits_{f\in\mathcal{P}(t)} NAV(f,t)}{\sum\limits_{f\in\mathcal{P}(t)} NAV(f,t) + Cash(\mathcal{P},t)} \tag{1}$$

where $\sum\limits_{f\in\mathcal{P}(t)} NAV(f,t)$ represents the sum of all Net Asset Value (NAV) for the underlying funds in the portfolio at period t. $Cash(\mathcal{P},t)$ accounts for the global uninvested cash in the portfolio, i.e., uncalled capital and possible distributions. Ideally, the investment degree ID should be as close as possible to 1. A trivial but not viable solution would be to bring $Cash(\mathcal{P},t)$ to 0 but this is without counting on future and inopportune capital calls exceeding the investor resources capacities. Becoming a defaulting investor once capital has been committed is subject to strong financial and reputational penalties. The PERP is therefore a challenging problematic for LP investors as they constantly need to stay close to the boundary without over-crossing it. In [18], authors modelled the problem as a sequence of single-period portfolio optimisation problems maximizing subsequent investment degrees using the following formulation:

$$\min_{C(\mathcal{P},t)} E_t\left[(1 - ID(\mathcal{P},t+1))^2\right] \tag{2}$$

where the $C(\mathcal{P},t)$ represents the optimal amount of capital to be recommitted at t. Note that this model only determines the optimal recommitment level with regards to the next period. This is debatable as the committed capital is called progressively over the investment period, i.e., roughly during the first 6 years. With respect to formulation (2), the optimal level of commitment at period t is therefore:

$$C(\mathcal{P},t) = E_t\left(\frac{Cash(\mathcal{P},t) + D(\mathcal{P},t+1) - \sum_{i=1}^{\tau}\gamma_{t+1,i+1}C_(\mathcal{P},t-i)}{\gamma_{t+1,1}}\right) \tag{3}$$

with E_t the conditional expectation, $Cash(\mathcal{P},t)$ the uninvested cash in the portfolio, $D(\mathcal{P},t)$ representing distributions for the next period, $C(\mathcal{P},t-i)$ the capital committed i period ago and $\gamma_{t+1,i+1}$ is the fraction of the capital committed i periods ago. $\gamma_{t+1,i+1}$ enables to compute the total capital called at the end of quarter $t+1$, i.e.,
$CC(\mathcal{P},t-i) = \sum\limits_{i=0}^{\tau}\gamma_{t+1,i+1}C(\mathcal{P},t-i)$ with τ representing the maximum fund age at which capital can still be called. Interested readers can refer to [18] for more details about the proof.

One can observe that the analytical solution requires to forecast distributions (see [16, 8]) at $t+1$ and the fraction of the capital committed in the past that will be called. Although prediction models can be developed to approximate future distributions, it is very unlikely to *guess* future capital calls as direct investments in private companies are made at the discretion of the fund's management.

Some works [18, 12] in the literature have tried to cope with this issue by engineering strategies using only available and past quantities. These strategies can be likened "heuristics" to approximate the optimal amount to be recommitted at each period and are defined as follows:

- $DZ^1(\mathcal{P}, t) = D(\mathcal{P}, t)$;
- $DZ^2(\mathcal{P}, t) = D(\mathcal{P}, t) + UC(\mathcal{P}, t - 24)$;
- $DZ^3(\mathcal{P}, t) = \frac{1}{ID(\mathcal{P},t)} \times (D(\mathcal{P}, t) + UC(\mathcal{P}, t - 24))$

Strategy $DZ^1(\mathcal{P}, t)$ recommits only current distributions at t while the strategy $DZ^2(\mathcal{P}, t)$ incorporates the uncalled capital made 24 quarters ago, i.e., $UC(\mathcal{P}, t - 24)$. The inclusion of this quantity is based on the observation that unallocated but committed capital for older funds that already passed their maximal NAV's peak is unlikely to be called. These funds are typically in the divestment period. The last strategy $DZ^3(\mathcal{P}, t)$ scales recommitments obtained from $DZ^2(\mathcal{P}, t)$ with the inverse of the current investment degree. If the investment degree is high, the recommitted capital will be decreased. Conversely, a low investment degree will amplify the recommitted capital. This allows to perform some kind of active control to adjust the level of recommitment to reach and remain stable at a target allocation .

In this paper, we propose to learn an active control system to recommit at each period. Instead of relying on cashflow predictions and strategies' engineering which require strong expert knowledge, we posit that recommitment policies could be learnt using a policy-based algorithm introduced in the next section.

4 Proximal Policy Optimisation

As aforementioned in section 2, the number of approaches relying on policy learning has flourished since recent years. They all try to find a trade-off between fast training and stability. Making large steps in the policy update can be disastrous, especially for on-policy algorithms which could never recover from subsequent updates. Among all existing alternatives in the literature, we considered the Proximal Policy Optimisation (PPO) algorithm [15] due to its simplicity. Although the PPO algorithm was released long after the Trust Region Policy Optimisation (TRPO) [13] which was the first of its kind, the PPO policy update is simpler but empirically seems to perform at least as well as TRPO relying on a second-order approach. But before diving into the stability improvement proposed in the PPO algorithm, let us recall the foundations, i.e., the vanilla policy gradient. Let π_θ represents a policy as a function of the parameter θ, the current state s_t, the taken action a_t and the received reward r_t at time t. A trajectory τ is a sequence of states and actions representing the path taken by an agent. In Reinforcement Learning, the goal is to discover the trajectory maximizing the expected return $J(\theta) = \mathbf{E}_{\pi_\theta}[R(\tau)]$ by updating sequentially the weights θ as follows: $\theta_{k+1} = \theta_k + \alpha * \nabla_\theta J(\theta_k)$ where $\nabla_\theta J(\theta_k)$ represents the policy gradient and is expressed as $\nabla_\theta J(\theta) = \mathbf{E}[R(\tau)\nabla_\theta \log \pi_\theta(a_t|s_t)]$. $R(\tau)$ can take different forms as suggested in [14]:

- the total reward trajectory: $\sum_{t=0} r_t$
- the future reward from action a_t or rewards-to-go: $\sum_{t=t'} r'_t$
- Future reward with baseline: $\sum_{t=t'} r'_t - b(s_t)$
- State-action value function: $Q^{\pi_\theta}(s_t, a_t)$
- Advantage function: $A^{\pi_\theta}(s_t, a_t) = Q^{\pi_\theta}(s_t, a_t) - V^{\pi}(s_t)$

All the previous choices lead to the same expected value but have different variance. The formulation using the advantage function is extremely common as it uses the state-action value function and the estimation value of the state as baseline to reduce the variance of the gradient. The PPO algorithm relies on an estimation of the advantage function and tries to avoid parameter updates that change the policy too much at one step. In the same way as TRPO, the loss function is built to measure of how policy π_θ performs relatively to an old policy $\pi_{\theta_{old}}$:

$$\mathcal{L}(\theta, \theta_{old}) = \mathbf{E}\left[A^{\pi_\theta}(s_t, a_t)\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\right] \tag{4}$$

While the TRPO algorithm uses the hard constraint $D_{KL}(\theta||\theta_{old}) < \lambda$ to limit the KL-divergence between both policies, the PPO algorithm relaxes the hard constraints and:

- either penalizes the KL-divergence directly in the loss function. This is the PPO-penalty version which we did not consider in this work.
- or clips the ratio $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ in the loss function to remove incentives for the new policy to get far from the old policy. Note that the KL-divergence is not used anymore as constraints nor as a penalty.

The PPO-clip algorithm considered in this work is depicted in Algorithm 1. Contrary to the penalty version in which penalty coefficients are adjusted automatically during training, PPO-clip requires a static hyper-parameter ϵ use to clip the ratio between the policies. Due to space restriction, we will not go further into details but more explanations can be obtained from the original paper [15].

5 Private Equity Recommitment as RL problem

As described in Section 3, the PERP can be solved using two main methodologies. While the first one relies on cashflow forecasting, the second one engineers recommitment functions only using past and current quantities from portfolios. Instead of building explicitly these functions, one could consider a Markov Decision Processes (MDP) to model a recommitment system and searches for the best policy in order to maintain a target investment degree while minimizing the risk of default.

Algorithm 1 PPO-clip version

1: Initialize policy parameters θ_1 and value function parameters ϕ_1
2: **for** $k \in \{1, ..., M\}$ **do**
3: Sample a set of trajectories $\{\tau_i\}_{i=1}^{M}$ using the policy π_{θ_k}
4: Create a batch $\mathcal{B}$ of transitions (s_t^i, a_t^i, r_t^i) $\forall t \in \{1, ..., |\tau_i|\}$ $\forall i \in \{1, ..., M\}$
5: Compute rewards-to-go $\hat{\mathcal{R}}_t^i$, i.e. rewards from action a_t^i, $\forall t \in \{1, ..., |\tau_i|\}$ $\forall i \in \{1, ..., M\}$
6: Estimate the advantages $A^{\pi_{\theta_k}}(s_t^i, a_t^i)$ using the value function V_{ϕ_k}
7: Perform policy update:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{M} \sum_{i=1}^{M} \frac{1}{|\tau_i|} \sum_{t=1}^{T_i} \left[\min \left(A^{\pi_\theta}(s_t^i, a_t^i) \frac{\pi_\theta(a_t^i|s_t^i)}{\pi_{\theta_{old}}(a_t^i|s_t^i)}, g\left(\epsilon, A^{\pi_\theta}(s_t^i, a_t^i)\right) \right) \right]$$

$$\text{with } g\left(\epsilon, A^{\pi_\theta}(s_t^i, a_t^i)\right) = \text{clip}\left(\frac{\pi_\theta(a_t^i|s_t^i)}{\pi_{\theta_{old}}(a_t^i|s_t^i)}, 1-\epsilon, 1+\epsilon \right)$$

8: Perform value function update by minimizing mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{M} \sum_{i=1}^{M} \frac{1}{|\tau_i|} \sum_{t=1}^{T_i} \left[V_\phi(s_t^i) - \hat{\mathcal{R}}_t^i \right]^2$$

9: **end for**

5.1 Modelling

Fig. 1 illustrates how the PERP can be turned into a Reinforcement Learning problem. Each state s_t represents the portfolio position at time t and contains the following information:

- $ID(\mathcal{P}, t)$: Investment degree at time t
- $D(\mathcal{P}, t)$: Distributions obtained from divestments at time t
- $CC(\mathcal{P}, t)$: Capital called at time t
- $UC(\mathcal{P}, t-24)$: Uncalled capital from commitment made 24 quarters ago
- $Cash(\mathcal{P}, t)$: Portfolio cash at time t
- $NAV(\mathcal{P}, t)$: Net Asset Value at time t

The state s_t gives us the opportunity to control the amount of recommitted capital at time t, i.e., the continuous action a_t depicted in Fig. 1. So far, the RL model is trivial to obtain. However, we need to be extremely cautious regarding the reward provided to the agent. Although we could define the reward by minimizing the deviation to the ideal investment degree as done in Equation 2, there is no control on the risk of default. Two alternatives open to us: (1) either we train on multiple portfolios per episode and adjust the objective using the standard deviation or (2) we constrain the agent to remain below the fateful boundary, i.e., $ID(\mathcal{P}, t) = 1.0$. Needless to say, alternative (2) is more challenging for the agent but we argue that it will be more generalizable than alternative (1). For this purpose, we define a local reward r_t^{valid} and a global reward r_τ^{ID}. While the former is applied after each action(recommitment), the second one only occurs at the end of a valid episode. We recall that a valid episode ends when the maximum number of steps has been reached. The agent is rewarded after each action depending on whether the future state of the portfolio is valid:

$$r_t^{valid} = \begin{cases} 0 \text{ if } ID(\mathcal{P}, t+1) > 1 \\ 1 \text{ if } else \end{cases}$$

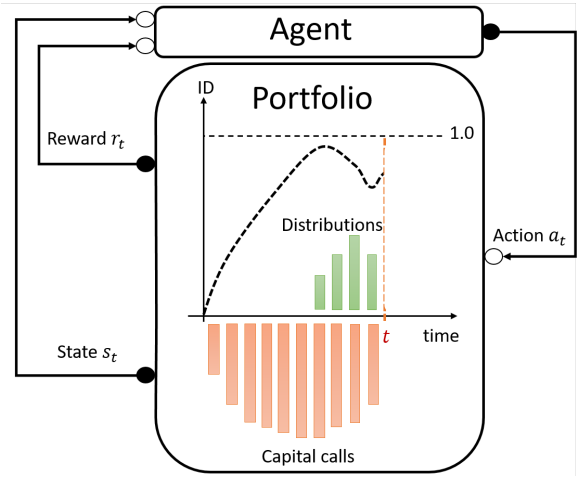


Fig. 1: Reinforcement Learning of private equity policies

If a situation of default happens, the episode is stopped and does not reach the maximum number of steps allowed. The accumulated reward obtained during the episode would finally correspond to the number of periods in which the portfolio remained valid. This reward function strictly increases monotonically to drive the agent to simply learn to provide valid episodes. Once the agent has learnt to recommit, i.e., it reaches the maximum number of steps per episode, it receives an additional and final global reward $r_\tau^{ID} = \sum_{t=1}^{T} ID(\mathcal{P}, t)$ where T is the maximal number of steps per episode. Note that the sum could be replaced by the min to maximize the worst investment degree obtained during an episode. Finally, the total reward of a valid episode is the accumulated local reward added to the shifted global reward:

$$r_\tau = r_\tau^{ID} \times 10^{(digits(T)+1)} + \sum_{t=1}^{T} r_t^{valid} \tag{5}$$

where $digits(T)$ is the number of digit of T. For an episode lasting 100 steps, $\#digit(100) = 2$. This shifting mechanism is a constraint handling approach to make sure that non-valid episodes are guaranteed to receive a total reward lower than valid ones.

5.2 Synthetic cashflows

Private equity data is a sensitive topic. Private equity players generally protect their rich cashflow histories. Although some financial data providers propose

commercial libraries for very specific periods and economies, their data are generally incomplete. Historical cashflows's data capture the fund's dynamics which is an essential information for training. Multiple works including [18] and [12] relied on commercial libraries to draw conclusions or train their own model. In this work, we adopt another strategy to simulate portfolio evolution over time. Since 1973, the Yale University's endowment has been investing in private equity using a methodology for modelling illiquid assets proposed by Takahashi and Alexander (see [16]). Referred to as the *mother of all cashflows's models*, this Yale-model can be applied to private equity and real asset funds (e.g. natural resources and infrastructures). Although, according to Takahasi and Alexander, the generated projections fit historical data, the cashflows are modelled as deterministic which limit their applicability.

Instead of depending on a commercial solution to acquire historical cashflows which are often expensive and incomplete, synthetic fund cashflows have been preferred in this work as they represent a more practical solution. This is the reason why we decide to rely on an alteration of the Yale-model to make it probabilistic. These synthetic cashflows are created by funnelling data generated by the robust and tried-and-tested, albeit over-simplistic, Yale-model through a noise-adding algorithm to construct a new dataset. The resulting dataset shows the statistical features and the useful patterns needed for capturing the liquidity risks associated with portfolio of funds. The synthetic cashflows considered in this work have been provided by T.Meyer, an expert in private equity and co-author of this paper.

6 Experimental setups

In order to fairly evaluate the resulting recommitment policies with the state of the art, simulations have been performed according to the parameters described in [18]. Due to the lack of secondary market, a portfolio cannot be bought instantaneously. We empirically created initial but mature portfolios over a year by committing equal capital to 16 randomly selected private equity funds. We also apply 30 % initial overcommitment in setting up all portfolios to be in line with the experiments performed in [18].

A portfolio simulation consists in recommitting some capital to new selected fund every quarter. The amount of capital is determined by the current policy sampled from the critic network (see Algorithm 1). Table 1a details the simulation parameters while Table 1b described the PPO-clip parameters. A single portfolio simulation last 104 quarters, i.e., 26 years. Capital is recommitted uniformly into 4 randomly selected funds. The number of portfolio simulations is therefore equal to the number of episodes:

$$\#episodes = \frac{steps\_per\_epoch \times epochs}{104} = 125000$$

Strategies $DZ^i(\mathcal{P}, t)$ for $i \in \{1, 2, 3\}$ proposed in [18] have been evaluated with the same parameters and over the same period. All experiments presented

in this paper were carried out using the HPC facility of the University of Luxembourg [17]. The python library SpinningUp [1] has been considered for the PPO-clip implementation. A distributed implementation using OpenMPI [5] has been considered to work with multiple environment in parallel. The discount parameter γ has been set to 1.0 since an episode's length is finite and last 26 years. The clip ratio ϵ has been set to 0.2 and represents how far can the new policy go from the old policy while still improving the objective. PPO-clip 's networks, i.e., actor and critic have both two hidden layers of 64 nodes. The ReLU function [2] has been chosen as activation function.

Table 1: Parameters

Parameters	**Training**	**Validation**
Cashflows frequency	quarterly	quarterly
Investment period	26 years	26 years
Funds per recommitment	4	4
Fund selection	random	random
Number of simulated portfolios	#episodes	1000

(a) Simulation parameters

Parameters	**Value**
steps_per_epoch	26000
gamma	1
epochs	500
# episodes	500
clip_ratio ϵ	0.2
pi_lr / vf_lr	$3e^{-4}$ / $1e^{-4}$
hidden layers	$[64, 64]$

(b) PPO-clip parameters

7 Experimental results

With regards to the experimental setups described in the previous section, Fig. 2 illustrates the average rewards recorded during policy optimisation/training. One can easily observe that the PPO-clip algorithm required few epochs to generate valid policies. The average rewards curve then steadily increases to reach what we can consider as a plateau in terms of improvements. Indeed, we can note periodic falls indicating that the algorithm have strong difficulties to improve more significantly the investment degree without breaking the cash constraint. When arrived at the rupture point, a policy yielding non-valid episodes is more likely to be generated leading to a steep fall in terms of overall rewards. When a fall occurs, the algorithm tries to recover until the next rupture. This pattern can be easily oberved in Fig. 2. Due to the shifting constraint handling approach implemented in this work, non-valid and valid episodes do not have the same reward scale which explains these deep reward falls every time the algorithm encounters a non-valid episode.

The best policy obtained after training is depicted in Fig. 2. In order to validate results, the obtained policy has been applied on a test set of 1000 portfolios. After recording the investment degree evolution and the validity of each portfolio, the average investment degree as well as the surrounding 95% confidence

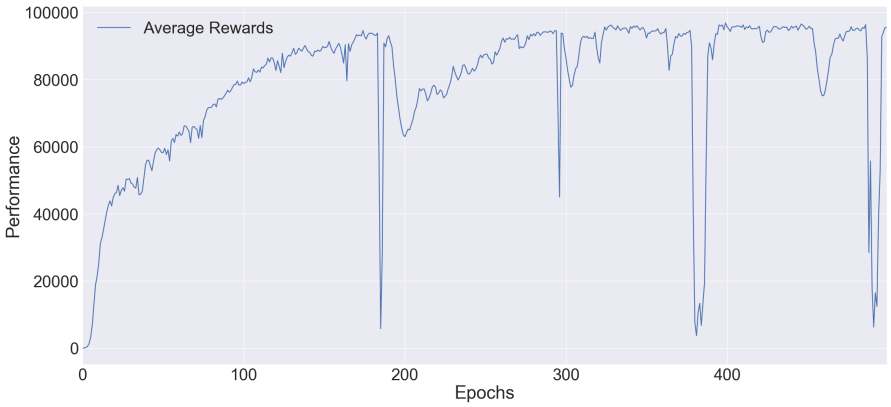


Fig. 2: Evolution of the average rewards per epoch

interval have been computed and are depicted in Fig. 3. We first observe that the percentage of overinvested portfolios remains extremely low, i.e. $\approx 0.7\%$. The investment degree varies strongly during the first 6 years going from 0.4 to almost 1.0. After the first 6 years, the average investment degree slightly increases to remain stable around 0.9.

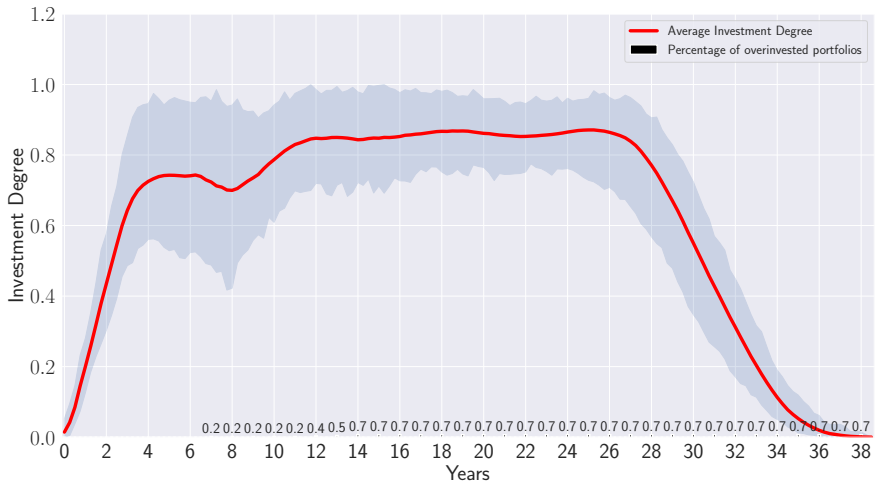


Fig. 3: Best policy obtained with the PPO-clip algorithm

We now compare the investment degree obtained with state-of-the-art strategies engineered in [18], namely DZ^i for all $i \in \{1, 2, 3\}$. Each DZ^i have been

applied on the same test set. Table 2 reports the average investment degree, the standard deviation of the investment degree and the fraction of overinvested portfolios obtained for each strategy including the best policy recommitment $PPO - clip^{best}$. Although the active recommitment period only lasts 26 years, we have still recorded the investment degree until portfolios were totally divested (38 years) to observe if there is no delay effect when applying a specific strategy. None of the 3 strategies have generated invalid portfolios. The investment degree reached by DZ^1 and DZ^2 remains low, i.e., below 0.6. Nevertheless, DZ^3 obtained the best results among the 3 strategies as reported in [18]. The recommitment policy $PPO - clip^{best}$ outperforms the 3 strategies by reaching a maximum investment degree above 0.8. Nonetheless, the DZ^3 reports better results during the first years as show in Fig. 4. The initial condition of the portfolio seems to be a challenge for the recommitment policy. Nevertheless, it is well-known in the literature that portfolio inception is a problem on its own. Therefore, we are not surprised by this under-performance at the beginning of the portfolio lifetime. In [18], authors discarded the first three years of the portfolio's lifetime to avoid the influence from the initial portfolio formation period.

Regarding the percentage of overinvested portfolios, it comes as no surprise to encounter some invalid portfolios when getting closer to $ID(\mathcal{P}, t) = 1.0$. This is due to cashflow variability which is very difficult to predict. An alternative would be to replace the strong cash constraint by a soft one taking the form of an additional objective. Most of the LP investors generally own multi-class asset portfolios. If liquidity is missing due to an unexpected capital calls, more liquid assets could be sold. Of course, such a situation should be tempered and the injected cash required to satisfy capital calls should be minimized. For this purpose, one could consider a multi-objective reinforcement learning algorithm.

8 Conclusion

Recommitting into new PE funds is crucial for LP investors to maintain high allocation to private equity. Current methodologies rely on cashflow forecasting and over-simplistic approaches which are lacking of flexibility. Although this problem is a key of major importance, few works have attempted to develop a robust and flexible recommitment system. Perhaps, this is due to the lack of data. This is the reason why we adopted a different strategy consisting in learning recommitment policies through Reinforcement Learning. Using synthetic cashflows build from the traditionnal but proven Yale-model, we applied Proximal Policy Optimisation to the Private Equity Recommitment Problem to maximise the investment degree while avoiding cash shortage situations by constraining the agent. Results obtained after training confirm that the recommitment policy outperform the strategies engineered in [18] while limiting the fractions of invalid portfolios. This work was a first proof of concept and subsequent experiments will be performed using different RL algorithms. Future works will investigate a strategy to handle the cash constraint more efficiently. Another avenue for research would be to model the cash constraint as a soft constraint, typically by

	$PPO - clip^{best}$			DZ^1			DZ^2			DZ^3		
years	mean	std	invalid (%)	mean	std	invalid (%)	mean	std	invalid (%)	mean	std	invalid (%)
0	0.07	0.02	0.00	0.07	0.02	0.0	0.07	0.02	0.0	0.07	0.02	0.0
1	0.29	0.03	0.00	0.29	0.03	0.0	0.29	0.03	0.0	0.30	0.03	0.0
2	0.52	0.04	0.00	0.52	0.04	0.0	0.52	0.04	0.0	0.55	0.03	0.0
3	0.68	0.06	0.00	0.69	0.04	0.0	0.69	0.04	0.0	0.75	0.03	0.0
4	0.73	0.06	0.00	0.75	0.04	0.0	0.75	0.04	0.0	0.83	0.03	0.0
5	0.74	0.07	0.00	0.76	0.04	0.0	0.76	0.04	0.0	0.85	0.04	0.0
6	0.74	0.07	0.08	0.71	0.05	0.0	0.71	0.05	0.0	0.81	0.05	0.0
7	0.71	0.08	0.20	0.63	0.05	0.0	0.63	0.05	0.0	0.74	0.05	0.0
8	0.71	0.07	0.20	0.56	0.04	0.0	0.57	0.05	0.0	0.70	0.04	0.0
9	0.75	0.05	0.20	0.54	0.03	0.0	0.56	0.03	0.0	0.72	0.04	0.0
10	0.80	0.05	0.20	0.56	0.03	0.0	0.58	0.03	0.0	0.76	0.03	0.0
11	0.84	0.05	0.23	0.58	0.02	0.0	0.60	0.02	0.0	0.79	0.03	0.0
12	0.85	0.05	0.40	0.59	0.02	0.0	0.62	0.02	0.0	0.81	0.03	0.0
13	0.85	0.05	0.58	0.59	0.02	0.0	0.62	0.02	0.0	0.81	0.03	0.0
14	0.84	0.06	0.70	0.58	0.02	0.0	0.60	0.02	0.0	0.79	0.03	0.0
15	0.85	0.06	0.70	0.56	0.02	0.0	0.58	0.02	0.0	0.77	0.03	0.0
16	0.85	0.06	0.70	0.55	0.02	0.0	0.57	0.02	0.0	0.76	0.03	0.0
17	0.86	0.06	0.70	0.54	0.02	0.0	0.57	0.02	0.0	0.76	0.03	0.0
18	0.86	0.07	0.70	0.55	0.02	0.0	0.58	0.02	0.0	0.77	0.02	0.0
19	0.86	0.07	0.70	0.55	0.02	0.0	0.58	0.02	0.0	0.78	0.02	0.0
20	0.85	0.07	0.70	0.55	0.02	0.0	0.58	0.02	0.0	0.79	0.02	0.0
21	0.85	0.08	0.70	0.55	0.02	0.0	0.58	0.02	0.0	0.78	0.02	0.0
22	0.85	0.08	0.70	0.54	0.02	0.0	0.58	0.02	0.0	0.78	0.02	0.0
23	0.85	0.08	0.70	0.54	0.02	0.0	0.57	0.02	0.0	0.77	0.02	0.0
24	0.86	0.08	0.70	0.54	0.02	0.0	0.57	0.02	0.0	0.77	0.02	0.0
25	0.86	0.08	0.70	0.54	0.02	0.0	0.57	0.02	0.0	0.77	0.02	0.0
26	0.85	0.08	0.70	0.54	0.02	0.0	0.57	0.02	0.0	0.78	0.02	0.0
27	0.81	0.09	0.70	0.53	0.02	0.0	0.56	0.02	0.0	0.76	0.02	0.0
28	0.73	0.08	0.70	0.49	0.02	0.0	0.52	0.02	0.0	0.71	0.03	0.0
29	0.62	0.08	0.70	0.44	0.02	0.0	0.46	0.02	0.0	0.62	0.03	0.0
30	0.50	0.07	0.70	0.37	0.02	0.0	0.39	0.02	0.0	0.51	0.03	0.0
31	0.38	0.06	0.70	0.29	0.02	0.0	0.31	0.02	0.0	0.40	0.03	0.0
32	0.27	0.05	0.70	0.21	0.02	0.0	0.22	0.02	0.0	0.29	0.03	0.0
33	0.17	0.04	0.70	0.14	0.02	0.0	0.14	0.02	0.0	0.19	0.03	0.0
34	0.09	0.02	0.70	0.07	0.01	0.0	0.08	0.01	0.0	0.10	0.02	0.0
35	0.04	0.01	0.70	0.03	0.01	0.0	0.03	0.01	0.0	0.05	0.01	0.0
36	0.01	0.01	0.70	0.01	0.01	0.0	0.01	0.01	0.0	0.02	0.01	0.0
37	0.00	0.00	0.70	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00	0.0
38	0.00	0.00	0.70	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00	0.0

Table 2: Summary statistics of the investment degree in recommitment strategies

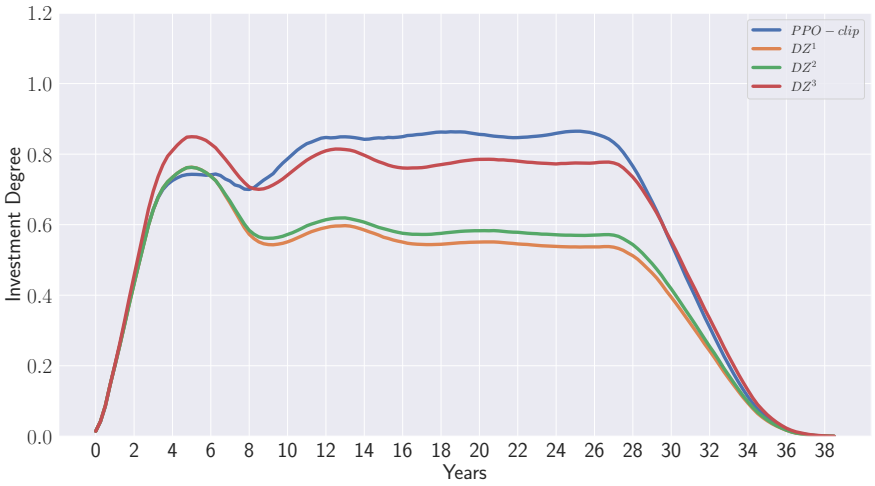


Fig. 4: Comparison between de Zwart's strategies [18] and the policy obtained with the PPO-clip algorithm

considering it as a second objective. Both opportunity cost and cash shortage are two conflicting objectives. Finally, this work could be extended to take into account multi-class asset portfolios.

Acknowledgment

E. Kieffer acknowledges the support of the European Investment Bank through its STAREBEI programme.

References

1. Achiam, J.: Spinning Up in Deep Reinforcement Learning (2018)
2. Agarap, A.F.: Deep learning using rectified linear units (relu) (2018), http://arxiv.org/abs/1803.08375, cite arxiv:1803.08375Comment: 7 pages, 11 figures, 9 tables
3. Arnold, T.R., Ling, D.C., Naranjo, A.: Waiting to be called: The impact of manager discretion and dry powder on private equity real estate returns. The Journal of Portfolio Management **43**(6), 23–43 (2017)
4. Cardie, J.H., Cattanach, K.A., Kelly, M.F.: How large should your commitment to private equity really be? The Journal of Wealth Management **3**(2), 39–45 (2000)
5. Gabriel, E., Fagg, G.E., Bosilca, G., Angskun, T., Dongarra, J.J., Squyres, J.M., Sahay, V., Kambadur, P., Barrett, B., Lumsdaine, A., Castain, R.H., Daniel, D.J., Graham, R.L., Woodall, T.S.: Open MPI: Goals, concept, and design of a next generation MPI implementation. In: Proceedings, 11th European PVM/MPI Users' Group Meeting. pp. 97–104. Budapest, Hungary (September 2004)

6. Hasselt, H.v., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. p. 2094–2100. AAAI'16, AAAI Press (2016)
7. Lerner, J., Schoar, A.: The illiquidity puzzle: theory and evidence from private equity. Journal of Financial Economics **72**(1), 3–40 (2004). https://doi.org/https://doi.org/10.1016/S0304-405X(03)00203-4, https://www.sciencedirect.com/science/article/pii/S0304405X03002034
8. de Malherbe, E.: Modeling private equity funds and private equity collateralised fund obligations. International Journal of Theoretical and Applied Finance **07**, 193–230 (2004)
9. Meyer, T.: Hidden in plain sight—the impact of undrawn commitments. The Journal of Alternative Investments **23**(2), 94–110 (2020)
10. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning (2013), http://arxiv.org/abs/1312.5602, cite arxiv:1312.5602Comment: NIPS Deep Learning Workshop 2013
11. Nevins, D., Conner, A., McIntire, G.: A portfolio management approach to determining private equity commitments. The Journal of Alternative Investments **6**(4), 32–46 (2004)
12. Oberli, A.: Private equity asset allocation: How to recommit? The Journal of Private Equity **18**(2), 9–22 (2015)
13. Schulman, J., Levine, S., Abbeel, P., Jordan, M.I., Moritz, P.: Trust region policy optimization. In: Bach, F.R., Blei, D.M. (eds.) ICML. JMLR Workshop and Conference Proceedings, vol. 37, pp. 1889–1897. JMLR.org (2015), http://dblp.uni-trier.de/db/conf/icml/icml2015.htmlSchulmanLAJM15
14. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. In: Proceedings of the International Conference on Learning Representations (ICLR) (2016)
15. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. CoRR **abs/1707.06347** (2017), http://dblp.uni-trier.de/db/journals/corr/corr1707.htmlSchulmanWDRK17
16. Takahashi, D., Alexander, S.: Illiquid alternative asset fund modeling. The Journal of Portfolio Management **28**(2), 90–100 (2002). https://doi.org/10.3905/jpm.2002.319836, https://jpm.pm-research.com/content/28/2/90
17. Varrette, S., Bouvry, P., Cartiaux, H., Georgatos, F.: Management of an academic hpc cluster: The ul experience. In: Proc. of the 2014 Intl. Conf. on High Performance Computing & Simulation (HPCS 2014). pp. 959–967. IEEE, Bologna, Italy (July 2014)
18. de Zwart, G., Frieser, B., van Dijk, D.: Private equity recommitment strategies for institutional investors. Financial Analysts Journal **68**(3), 81–99 (2012)

Heuristic Coordination in Cooperative Multi-Agent Reinforcement Learning

Ramon Petri[1], Eugenio Bargiacchi[2],
Huib Aldewereld[1], and Diederik M. Roijers[1,2]

[1] HU University of Applied Sciences, Utrecht, The Netherlands
ramon.petri@student.hu.nl &
{huib.aldewereld,diederik.yamamoto-roijers}@hu.nl
[2] Vrije Universiteit Brussel, Brussels, Belgium
{eugenio.bargiacchi,diederik.roijers}@vub.be

Abstract. Key to reinforcement learning in multi-agent systems is the ability to exploit the fact that agents only directly influence only a small subset of the other agents. Such *loose couplings* are often modelled using a graphical model: a coordination graph. Finding an (approximately) optimal joint action for a given coordination graph is therefore a central subroutine in cooperative multi-agent reinforcement learning (MARL). Much research in MARL focuses on how to gradually update the parameters of the coordination graph, whilst leaving the solving of the coordination graph up to a known typically exact and generic subroutine. However, exact methods – e.g., Variable Elimination – do not scale well, and generic methods do not exploit the MARL setting of gradually updating a coordination graph and recomputing the joint action to select. In this paper, we examine what happens if we use a heuristic method, i.e., local search, to select joint actions in MARL, and whether we can use outcome of this local search from a previous time-step to speed up and improve local search. We show empirically that by using local search, we can scale up to many agents and complex coordination graphs, and that by reusing joint actions from the previous time-step to initialise local search, we can both improve the quality of the joint actions found and the speed with which these joint actions are found.

Keywords: Coordination Graphs · Local Search · Multi-agent Reinforcement Learning · Multi-agent Thompson Sampling

1 Introduction

Coordination is an important aspect of everyday life – whether playing football, or participating in traffic. In artificial intelligence, coordination between multiple artificial agents is therefore a popular topic, with applications ranging from robotic rescue operations [Visser et al., 2014, Chalup et al., 2019] to maintenance scheduling on highways between multiple contractors [Scharpff et al., 2016, Scharpff, 2020].

Key to keeping cooperative multi-agent coordination tractable is to exploit so-called *loose couplings*, i.e., the property that individual agents typically only *directly* affect a small subset of the other agents. For example, imagine a wind farm, where each agent controls the yaw of a wind turbines [Verstraeten, 2021]. A turbine produces turbulence in its wake, which can affect the wind turbines behind it. The turbines behind the first one may in turn affect other wind turbines, ultimately still requiring coordination between the entire wind farm, but the first turbine only directly affects the ones behind it. Such loose couplings can be expressed using a graphical reward structure called a *coordination graph* [Verstraeten et al., 2021]. In coordination graphs, direct influence between agents is modelled as a local reward function, that has the joint action space of the agents affecting and being affected as its domain.

When the coordination graphs and all its local reward functions are known, the optimal joint action can be found using non-serial dynamic programming [Bertele and Brioschi, 1972], or as it is more commonly known in the agents community, *variable elimination (VE)* [Guestrin et al., 2002]. Variable elimination is an exact algorithm [Rosenthal, 1977], but due to the inherent hardness of solving coordination graphs, scales poorly in the connectivity of a graph, and typically also in the number of agents.

When the ground truth local reward functions in a coordination graph are unknown to the agents, we are in the multi-objective multi-armed bandit setting [Verstraeten et al., 2020]. In this setting, the local reward functions must be learned through interaction with the environment (e.g., the wind farm) [Bargiacchi et al., 2018]. A state-of-the-art algorithm for doing so is called *multi-agent Thompson sampling (MATS)* [Verstraeten et al., 2020]. MATS uses VE as a subroutine to find the optimal joint action for the graphs sampled before each interaction with the environment. As such, MATS scales poorly in the connectivity of the coordination graphs, and typically also the number of agents in the graph.

In this paper, we study the effect of using *local search (LS)* [Russell and Norvig, 2005] algorithms as a subroutine in multi-agent Thompson sampling. These heuristic algorithms scale extremely well in the number of agents and the connectivity of the coordination graphs, but they are of course not exact. We can therefore expect to incur more *regret* [Verstraeten et al., 2020], i.e., a larger cumulative difference between the optimal team rewards and the rewards resulting from the joint actions that are performed, then when using VE. We do however expect a large gain in runtime, especially for increasingly complex coordination graph.

We observe that when we use LS as a subroutine inside of MATS, there are some aspects that we can exploit. Firstly, as the newly obtained information obtained at each timestep has a relatively smaller impact on the posterior beliefs over the true mean rewards, the sampled coordination graphs at each timestep are increasingly similar. Secondly, at each timestep, we produce a joint action to execute using LS. Finally, local search algorithms can benefit from initialization with a good initial solution, i.e., an educated guess for a joint action. We therefore

propose to reuse the joint action found by LS at the previous timestep as the starting point for LS in the next timestep. Using this together with an iterative search scheme, this leads to our *reusing iterative local search (RILS)* algorithm. We show experimentally that RILS is able to find good approximate solutions for coordination graphs. When used in MATS, this leads to higher regret, but at a fraction of the runtime of MATS with VE as a subroutine. Furthermore, the difference in regret becomes smaller as the coordination graphs become more complex, while the difference in runtime becomes larger. We therefore conclude that RILS is a suitable algorithm to scale up to complex coordination graphs in multi-agent multi-armed bandits.

2 Background

A *coordination graph (CoG)* models the (sparse) relationships between multiple cooperative agents. In a coordination graph, each agent is represented by an individual node. An edge between two nodes indicates that coordination between their associated agents is required to achieve optimal behaviour. We note that while edges in coordination graphs are often represented in a pairwise fashion, we use a hyper-edge representation, where each edge can connects several agents at once [Roijers et al., 2015b]. Each hyper-edge is associated to a *local reward function*, which specifies the rewards for a subset of agents.

In planning, the local reward functions specify a deterministic (or expected) local reward given a local joint action by the connected agents [Roijers, 2016]. In this paper, however, we are concerned with a reinforcement learning setting in which the local reward functions are stochastic, and specified using a distribution over local rewards. This is called a multi-agent multi-armed bandit (MAMAB) [Bargiacchi et al., 2018]. More formally, a multi-agent multi-armed bandit (MAMAB), is a tuple $\langle D, A, f \rangle$ where:

- D is the set of all m agents.
- $A = A_1 \times \ldots \times A_m$ is the joint action space.
- $f : A \to \mathbf{R}$ is the global reward function, i.e. a random function[3] associating each full joint action to a sampled reward. In a MAMAB, f can be decomposed into a set of ρ independent local reward functions, such that $f(a) = \sum_{e=1}^{\rho} f^e(a^e)$. Note that each of these constituent components are again random functions.

Intuitively, it should be possible to exploit the structure of f and the coordination graph to quickly learn the local reward functions, without incurring in an exponential regret from the large full joint-action space of a multi-agent setting (the curse of dimensionality). Multi-agent Thompson Sampling (MATS) [Verstraeten et al., 2020] is an algorithm that does precisely this: it maintains a

[3] A random function is the function equivalent of a random variable, i.e., a function which is defined in terms of an experiment of which the outcome varies according to a given probability distribution. As such evaluating a random function for the same input twice may yield a different output.

posterior distribution of the mean rewards for each possible local joint action, which it samples when it needs to act in the MAMAB. Such a sample leads to a coordination graph with non-stochastic rewards. The full joint action selected is then the one that maximizes the reward across all sampled local arms. This strategy provably [Verstraeten et al., 2020] results in a regret that is linear in the number of agents, rather than exponential.

In order to select the best joint action, MATS must maximize across all local arms in a computationally efficient manner. To do this, MATS relies on an exact algorithm that was originally devised to marginalize discrete variables in probabilistic graphical models. This is not surprising, as the concept of a coordination graph is analogous to an undirected graphical model.

In particular, MATS uses a well-known algorithm called Variable Elimination (VE) [Bertele and Brioschi, 1972, Rosenthal, 1977, Guestrin et al., 2002]. Originally developed to perform exact inference, VE can be used to determine the optimal action of multiple agents maximizing a factored reward function, in our case f. VE is an iterative algorithm, which progressively removes each agent from the coordination graph after computing its best response w.r.t. its neighbors, i.e., for each local joint action of the neighbors it determines the action that maximizes the total reward of the group. The advantage of VE over naive brute force search is in its computational complexity, which is combinatorial on the induced width of the graph, i.e., the largest local action space considered during the elimination process.

The computational complexity of VE for sparse coordination graphs is much lower than naive brute forcing, which is exponential in the number of agents. However, VE still tends to perform poorly when dealing with large number of agents, as the induced width typically increases with the number of agents, albeit much slower than the number of agents itself.In turn, this prevents using the MATS algorithm in large scale bandits, unless VE is replaced by an approximate selection technique.

A popular approximate optimization algorithm that is called Iterative Local Search (ILS). This technique is based on Local Search (LS) as described by [Russell and Norvig, 2005] and has an extension in the form of an iterative variant (ILS) as described by [Lourenço et al., 2003]. Local search algorithms take an optimisation problem – such as a coordination graph – and find approximate solutions by starting with a random solution, and looking in the neighborhood of the current solution, as defined by a set of allowed small mutations, for improvements. Iteratively applying such improvements until no improvements can be found in the neighborhood leads to a so-called local optimum. ILS can escape such local optima, by performing larger random mutations and re-applying local search.

3 Algorithms

In this paper we investigate the potential of applying *local search* and *iterative local search* schemes as an approximate subroutine in MATS [Verstraeten et al.,

2020] to replace the exact VE subroutine. First, we define how local search can be applied to coordination graphs. Secondly, we create an algorithm that performs iterative local search while exploiting the *reinforcement learning* setting. Specifically, at each timestep, the MATS algorithm learns more about the local rewards functions, and updates its local posterior mean reward distributions, before re-sampling a coordination graph to select the next timestep's joint action (using the VE or local search subroutines). We make the following observations about the learning process of MATS using (iterative) LS as a subroutine:

- The sampled coordination graphs at each timestep are increasingly similar. This is because the new information gathered at each timestep has a diminishing impact on the posterior mean reward distributions with respect to the information already gathered. Morevoer, over time, the posterior mean reward distributions become increasingly certain about what the local mean rewards ought to be, leading to narrower distributions and therefore more similar samples.
- At each timestep, we produce a joint action to execute.
- Local search algorithms can benefit from initialization with a good initial solution.

Combining these observations, we observe that it is likely beneficial to reuse the joint action found and executed at the previous timestep as the initial starting solution for (iterative) local search in the current timestep. We propose an algorithm - the Reusing Iterative Local Search algorithm (RILS) - that does so. We thus exploit the multi-agent reinforement learning setting to speed up our heuristic search subroutine.

3.1 Local Search for Coordination Graphs

The Local Search (LS) algorithm for coordination graphs (Algorithm 1) works by incrementally updating a joint action with local improvements, i.e., changes in actions for a single agent that improve the global reward. Starting from a random joint action (an array of individual actions for each agent), ar, the algorithm goes through all the agents in the graph in random order and for each agent, v, through all the actions, a available to that agent, to check whether replacing $ar[v]$ with a yields an improvement. We note that to do so, the algorithm only needs to calculate the difference, Δ, in reward for the sum of local reward functions that have agent v in scope. This therefore takes only a fraction of the time of a full evaluation of ar over the entire graph. If that Δ is bigger than zero, i.e., a is an improvement upon the current action of agent v, $ar[v]$ is changed to a. The algorithm uses a flag *changed* to check whether the last pass over the agents yielded an improvement; it is set to continue the while loop until no higher rewards can be found. When no local improvements upon ar can be found by updating the action of any of the agents in the coordination graph, LS has converged to a local optimum and the total team reward is returned. The total team reward is evaluated by a function named $evalTeamReward$ that sums

over all the local reward functions. This makes $evalTeamReward$ an expensive operation, and it is therefore key to the performance of LS as a subroutine within MATS that this happens only once per call to LS.

Algorithm 1 Local Search

```
1: procedure LOCAL SEARCH(startAction)
2:     ar ← startAction or a random joint action if startAction is null
3:     changed ← true
4:     while changed do
5:         changed ← false
6:         for v ∈ agents do                ▷ In a different random order each iteration.
7:             for a ∈ actions[v] do
8:                 Δ ← evaluateLocalActionChange(ar, v, a)
9:                 if Δ > 0 then
10:                    ar[v] ← a
11:                    changed ← true
12:                end if
13:            end for
14:        end for
15:    end while
16:    return (ar, evalTeamReward(ar))
```

LS can be used by itself as a subroutine within MATS. In this case, LS then starts from a random joint action each time that it is called. While this is no doubt a highly efficient heuristic, it lacks in two key aspects: the local optima achieved do not converge to the optimal joint action over time, due to the random nature of LS, and, in the multi-agent reinforcement learning setting, we start anew at each timestep to select a joint action, disregarding the information about how good the joint action that LS found on the previous timestep was.

3.2 Reusing Iterative Local Search (RILS)

Iterative Local Search (ILS) uses Local Search as a subroutine to escape local optima, by making larger randomized changes, called perturbations, to the solution after LS runs into a local optimum and then rerunning LS to see whether this leads to further improvements. Note that as this can in principle continue indefinitely, typically a maximum number of iterations is set to limit the number of trials in which ILS can maximize its result.

The added randomization uses a so called perturbation probability (PP), i.e., with a probability PP each part of the solution is set to a random value. For coordination graphs we employ local-reward-function-based perturbations, which means that we iterate over all local reward functions, and with probability PP, the actions for all agents in scope of the reward function are changed to a random action. We chose this over an agent-based perturbation strategy, because

Algorithm 2 RILS

```
1: procedure Reusing Iterative Local Search(numOfTrials, PP, PRandom, previousAction)
2:     if previousAction = Empty ∨ rn < PRandom then
3:         ar ← randomAction()
4:     else
5:         ar ← previousAction
6:     end if
7:     val ← evalTeamReward(ar)
8:     for i ← 0 to numOfTrials do
9:         ac ← ar
10:        rn ← randomnumber ∈ [0, 1]
11:        if previousAction = Empty ∨ rn < PRandom then
12:            ac ← randomAction()
13:        else
14:        for Each local reward function f^e do
15:            rn ← randomnumber ∈ [0, 1]
16:            if rn < PP then
17:                Change actions of all agents in scope of f^e to a random action in ac.
18:            end if
19:        end for
20:        ac′, val′ ← LS(ac)                                   ▷ Algorithm 1
21:        if val′ > val then
22:            ar ← ac′
23:            val ← val′
24:        end if
25:    end for
26:    previousAction ← ar
27:    return ar, val
```

if the action of a single agent changes, without any actions of its neighbours changing, LS will change this action straight back towards the previously found local optimum.[4]

In order to exploit the multi-agent reinforcement learning setting in the MATS algorithm, we propose Reusing Iterative Local Search (RILS) (Algorithm 2). The algorithm works by checking if a *previousAction* is available, from the previous iteration of the MATS algorithm. If not (i.e., this is the first timestep of MATS), the current joint action ar gets initialized with a random joint action. If it is available, ar is initialized with the *previousAction*. Subsequently, ar is evaluated to get its team reward, val. Note that it is necessary to re-evaluate the previous joint action between timesteps, as MATS samples the local reward functions each timestep, leading to slightly different local rewards.

[4] In order to make sure, we did in fact try out the agent-based perturbation strategy as well, but this indeed proved far less effective, so for the remainder of this paper we only use the local-reward-function-based perturbation strategy.

The main loop of RILS runs through a number of trials, $numOfTrials$, to try and find better solutions than the previous joint action (or the randomly generated one). This is done by first perturbating ac using the previously described local-reward-function-based perturbation strategy, after which it gets passed to Algorithm 1: LS. If the new local optimum found by LS, ac' with value val' improves over the previous ar, this joint action ac' replaces the current best, ar.

While iteratively improving upon the same joint action can be effective, and can save a lot of runtime due to efficient initialisation, there is also a risk. Specifically, RILS might get stuck in the same local optimum for a very long time, especially if that local optimum turns out to be hard to escape by small random perturbations. Therefore, RILS also has a very small probability, $PRandom$, to start from a completely random solution at the beginning of its main loop.

When the number of trials are up, RILS stores the best found joint action, ar in $previousAction$, and returns it along with its team reward, val.

4 Experiments

We now compare *Local Search (LS, Algorithm 1)* and *Reusing Iterative Local Search (RILS, Algorithm 2)* against Variable Elimination (VE) [Guestrin et al., 2002] as a subroutine within the Multi-Agent Thompson Sampling (MATS) algorithm [Verstraeten et al., 2020], both in terms of regret and in terms of runtime, for increasingly complex MOMABs. We use the implementations of VE and MATS found in the AI-Toolbox [Bargiacchi et al., 2020]. Additionally, our implementation of LS will be released in the same toolbox.

Our experiments are based on the Gem Mining problem from [Bargiacchi et al., 2018, Verstraeten et al., 2020], which is adapted from the Mining Day problem from [Roijers et al., 2015b], which is a multi-objective coordination graph benchmark problem. Gem Mining is engineered in such a way that the induced width – the primary indicator for the complexity of a coordination graph – can be controlled without changing the number of agents.

In Gem Mining, a mining company mines gems from a set of mines (local reward functions) located in the mountains (see Figure 1). The mine workers live in villages at the foot of the mountains. The company has one van in each village (agents) for transporting workers and must determine every morning to which mine each van should go (actions), but vans can only travel to nearby mines (graph connectivity). Workers are more efficient when there are more workers at a mine: the probability of finding a gem in a mine is $x \cdot 1.03^{w-1}$, where x is the base probability of finding a gem in a mine and w is the number of workers at the mine. To generate an instance with v villages (agents), we randomly assign 1-5 workers to each village and connect it to a between y and z mines. Each village is only connected to mines with a greater or equal index, i.e., if village i is connected to m mines, it is connected to mines i to $i + m - 1$. The last village is connected to z mines and thus the number of mines is $v + z - 1$.

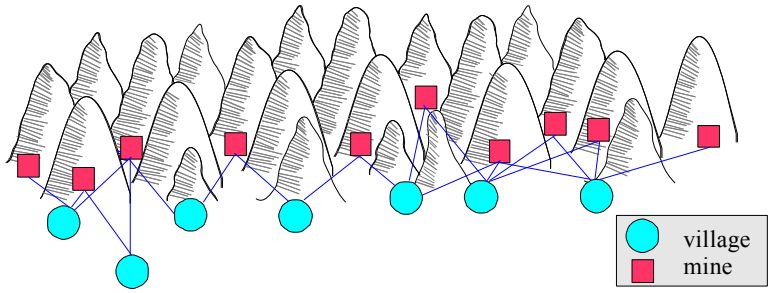


Fig. 1. Gem Mining example. Each village represents an agent, while the mines represent the local reward functions.

4.1 Gem Mining Problem

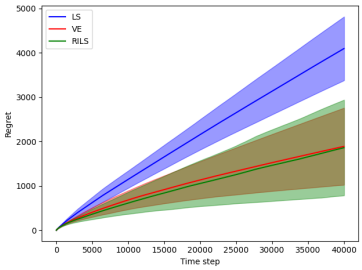
The Gem Mining problem is so constructed that the induced width is limited. This induced width, is the number of neighboring agents (villages) at the time the agent (village) is eliminated by the variable elimination (VE) algorithm [Guestrin et al., 2002].

The induced width can be used as a measure of the complexity of the graph. Due to the chain-like shape of the Gem Mining problem, the elimination order for VE can always be chosen to be from left to right (or the reverse) with minimal resulting induced width. Specifically, as the left-most mine in the graph always has a maximal number of neighbours z, and eliminating that agent does not increase the the number of neighbours of the subsequent villages, the induced width of a Gem Mining problem is always z. The Gem Mining problem is therefore well-suited to show how algorithms behave when the graph complexity increases.

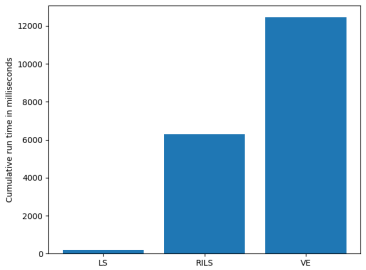
4.2 Results

We run MATS using LS, RILS and VE as a subroutine on the same randomly generated Gem Mining instances of 20 agents (villages), with varying induced width, i.e., between 3 and 6. For each induced width level, we perform 20 runs. For RILS, we use $numOfTrials = 15$, $PP = 0.001$, and $PRandom = 0.0001$, for each experiment. All experiments were performed on a TOXIC-15CL872-1060 customised BTO laptop, with 16.0 GB ram and a processor intel core I7-8750H of 2.20 GHz and 6 cores

For 1-3 villaged per mine, and a resulting induced width of 3, we observe in Figure 2(a) that the cumulative regret of MATS while using LS as a subroutine is significantly higher than that of MATS with VE or RILS. However, MATS with LS uses only a fraction of the runtime (Figure 2(b)) of MATS with VE or RILS. VE uses the most runtime, with RILS using about half the runtime of VE. While MATS with VE and MATS with RILS (with 15 trials) reach about the



(a) Regret vs, timesteps



(b) Run run time in milliseconds VE, HULSA and IHULSA

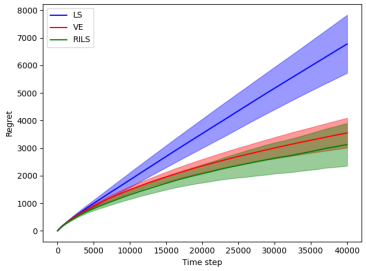
Fig. 2. Results when running MATS with a graph size of 20 villages and 1-3 mines per village

same regret, MATS using RILS has a slightly higher variance in its regret than when using VE. This is expected as RILS is a randomised heuristic algorithm.

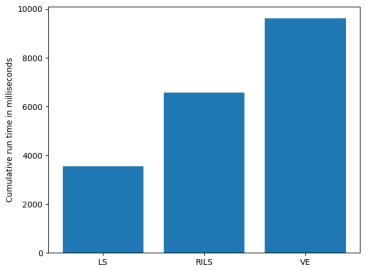
As the induced with increased we observe interesting patterns in regret (Figures 3(a)–5(a)). Firstly, the regret of using LS gets closer to that of using VE and RILS. This is probably because, as the number of neighbouring agents per agent increases, there are possibilities for gradual improvements for an hill-climbing algorithm like LS, i.e., it takes longer to run into a local optimum. Secondly, the regret of using RILS seems to dive under the regret of using VE. This can probably be explained by its reuse – even though the graph sampling of MATS might lead to a new joint action, the joint action of the previous timestep, as reused by RILS, may very well still be a local optimal. Therefore, while VE is guaranteed to follow the exploration mechanism of MATS, RILS is not. While this may lead to better in practice performance, this lack of exploration does break the regret guarantees of MATS [Verstraeten et al., 2020]. For the intent of this paper however, we are mainly interested in scalability and performance.

In terms of runtime (Figures 3(b)–5(b)), we observe a different pattern. Firstly, the runtimes of LS and RILS do increase with the complexity of the graphs. This can be explained from the observation that in some complex graphs it also takes longer to find a local optimum (even though the quality of that local optimum is likely to be higher). However, ultimately, as can be seen in Figure 5(b), for ever more complex graphs, VE has a much larger increase in runtime than LS and RILS.

Another key observation in terms of runtime is that for low induced width (Figure 2(b), LS has a much lower runtime than RILS. RILS uses 15 trials to find new local optima, and its efficiency gain due to reuse is clearly not able to compensate for the multiple trials yet. However, as the induced width increases, and finding a local optimum from a completely random solution takes more time, the runtime of MATS with LS overtakes the runtime of MATS with RILS, even if RILS is using 15 trials instead of the 1 for LS. This indicated that reuse is being effective; the reused initial joint action (i.e., the best joint action found

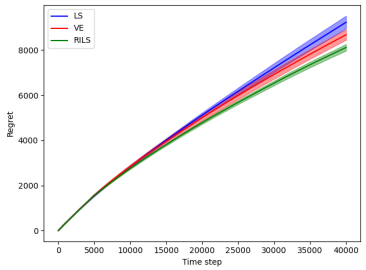


(a) Regret vs. timesteps

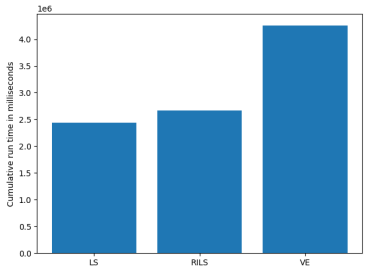


(b) Runtime in milliseconds for MATS in combination with VE, LS and RILS

Fig. 3. Results when running MATS with a graph size of 20 villages and 2-4 mines per village

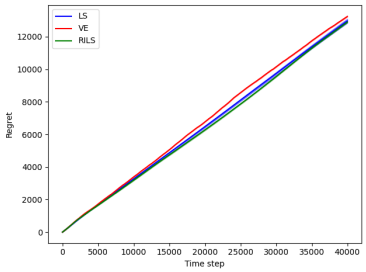


(a) Regret vs. timesteps

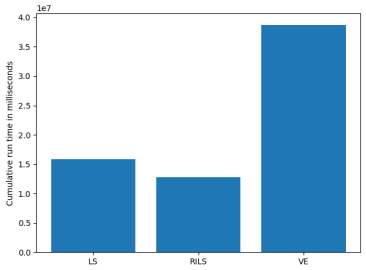


(b) Runtime in milliseconds for MATS in combination with VE, LS and RILS

Fig. 4. Results when running MATS with a graph size of 20 villages and a 3-5 mines per village



(a) Regret vs, timesteps



(b) Runtime in milliseconds for MATS in combination with VE, LS and RILS

Fig. 5. Results when running MATS with a graph size of 20 villages and 4-6 mines per village

for the previous timestep), is much closer to the ultimately selected joint action, and therefore takes considerably less time to find.

We therefore conclude that even though MATS using RILS as a subroutine does lose its theoretical regret bounds due to the heuristic nature of the RILS algorithm, the in practice regret can be good, whilst scaling much better in the complexity, i.e., induced width, of the graphs than VE, and even LS.

5 Related work

In this paper, we have proposed RILS – an approximate subroutine for optimising the joint action in coordination graphs for multi-agent reinforcement learning in a multi-agent multi-armed bandit (MOMAB) setting. For this setting we have integrated RILS with MATS [Verstraeten et al., 2020], the state-of-the-art in MOMABs. Other algorithms also apply to this setting however, and RILS can be used in those algorithms as well. For example, sparse cooperative Q-learning [Kok and Vlassis, 2004, Kok and Vlassis, 2006, Bargiacchi et al., 2018, Verstraeten et al., 2020] can be used in this setting, and RILS can directly replace the joint action selector subroutine there as well. Furthermore, RILS can also be easily adapted for usage in multi-agent upper confidence exploration (MAUCE) [Bargiacchi et al., 2018]. Specifically, as MAUCE keeps vector-valued rewards, and uses an adapted variant of VE that scalarises these to determine which vector is, RILS also should keep vector-valued rewards, and be aware of the value of the whole joint action to determine whether the difference in vector-valued rewards while running LS (Algorithm 1) are indeed improvements.

We note that other than local-search based algorithms. There are also other classes of approximate algorithms that seem promising, such as, a.o., Max-plus [Kok and Vlassis, 2005], AND/OR tree search methods [Marinescu and Dechter, 2005], variational methods [Liu and Ihler, 2013, Roijers et al., 2015a]. We note though that these have not been adapted for the multi-agent RL in MOMABs, and it would be interesting to investigate whether reuse schemes that exploit information from the previous iteration work for those algorithms as well. There may also be potential to use different initialisation schemes that leverage previous observations from interaction with the environment as well. For example, one may consider deep learning for coordination graphs [Böhmer et al., 2020], in order to determine the initial solution before running local search.

Finally, we note that this work may be extended to use in factored or multi-agent MDP settings [Boutilier, 1996]. In such settings, the coordination graph would depend not only on the actions of the agents, but also on state variables, that are provided by the environment. Therefore, multi-agent RL algorithms for this setting (e.g., [Kok and Vlassis, 2004, Kok and Vlassis, 2006, Bargiacchi et al., 2021]) are faced with different coordination graphs at every timestep, but can still use subroutines like VE to find the joint actions. In this context, RILS would have to be adapted, e.g., by finding the last joint action for the most similar state previously observed. Initialisation using deep learning [Böhmer et al., 2020], might be especially promising in this context.

6 Conclusion

In this paper, we proposed the heuristic *reusing iterative local search (RILS)* algorithm, as an alternative to exact joint action finders for multi-agent cooperative reinforcement learning in MOMABs, and specifically in combination with the multi-agent Thompson sampling (MATS) [Verstraeten et al., 2020] algorithm. RILS reuses the joint action found at the previous timestep to initialise its search for a new joint action. This is effective as, as the information accrued through interaction with the environment accumulates, the new information gained at each timestep impacts the learned reward structure (i.e., coordination graph) for the next timestep less and less. This makes the graphs for subsequent timesteps increasingly similar, and therefore the joint action of the previous timestep increasingly likely to be a good initialisation. We have shown experimentally that using RILS is able to closely match the regret for an exact subroutine, while using significantly less runtime. Moreover, its runtime scales better in the complexity of the graphs. We therefore believe RILS can be key to keep multi-agent reinforcement learning in MOMABs scalable for complex graphs.

In future work, we aim to investigate the combination of RILS with different algorithms such as sparse cooperative Q-learning [Kok and Vlassis, 2006] and MAUCE [Bargiacchi et al., 2018]. Furthermore, we aim to investigate larger, and real-world inspired problems, such as wind farms [Verstraeten et al., 2021]. Finally, we aim to investigate how a reusing iterative local search scheme can be applied in reinforcement learning in multi-agent Markov decision processes (MMDPs) [Boutilier, 1996], and multi-objective multi-agent reinforcement learning settings [Rădulescu et al., 2020].

Acknowledgements

The authors would like to acknowledge FWO (Fonds Wetenschappelijk Onderzoek) for their support through the SB grant of Eugenio Bargiacchi (#1SA2820N). This research was supported by funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme. This project was supported by the KIEM project "Exploration towards an intelligent Scoliosis brace" that was funded by the Stichting Innovatie Alliantie in the Netherlands.

References

[Bargiacchi et al., 2020] Bargiacchi, E., Roijers, D. M., and Nowé, A. (2020). AI-Toolbox: A C++ library for reinforcement learning and planning (with python bindings). *Journal of Machine Learning Research*, 21(102):1–12.

[Bargiacchi et al., 2018] Bargiacchi, E., Verstraeten, T., Roijers, D., Nowé, A., and Hasselt, H. (2018). Learning to coordinate with coordination graphs in repeated single-stage multi-agent decision problems. In *International conference on machine learning*, pages 482–490. PMLR.

[Bargiacchi et al., 2021] Bargiacchi, E., Verstraeten, T., and Roijers, D. M. (2021). Cooperative prioritized sweeping. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 160–168.

[Bertele and Brioschi, 1972] Bertele, U. and Brioschi, F. (1972). *Nonserial dynamic Programming*. Academic Press, N.Y.

[Böhmer et al., 2020] Böhmer, W., Kurin, V., and Whiteson, S. (2020). Deep coordination graphs. In *International Conference on Machine Learning*, pages 980–991. PMLR.

[Boutilier, 1996] Boutilier, C. (1996). Planning, learning and coordination in multiagent decision processes. In *TARK*, volume 96, pages 195–210. Citeseer.

[Chalup et al., 2019] Chalup, S., Niemueller, T., Suthakorn, J., and Williams, M.-A., editors (2019). *RoboCup 2019: Robot World Cup XXIII*, Lecture Notes in Artificial Intelligence, Berlin, Germany. Springer.

[Guestrin et al., 2002] Guestrin, C., Lagoudakis, M., and Parr, R. (2002). Coordinated reinforcement learning. In *ICML*, volume 2, pages 227–234. Citeseer.

[Kok and Vlassis, 2004] Kok, J. R. and Vlassis, N. (2004). Sparse cooperative Q-learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 61.

[Kok and Vlassis, 2005] Kok, J. R. and Vlassis, N. (2005). Using the max-plus algorithm for multiagent decision making in coordination graphs. In *Robot Soccer World Cup*, pages 1–12. Springer.

[Kok and Vlassis, 2006] Kok, J. R. and Vlassis, N. (2006). Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, 7:1789–1828.

[Liu and Ihler, 2013] Liu, Q. and Ihler, A. (2013). Variational algorithms for marginal MAP. *The Journal of Machine Learning Research*, 14(1):3165–3200.

[Lourenço et al., 2003] Lourenço, H. R., Martin, O. C., and Stützle, T. (2003). Iterated local search. In *Handbook of metaheuristics*, pages 320–353. Springer.

[Marinescu and Dechter, 2005] Marinescu, R. and Dechter, R. (2005). AND/OR branch-and-bound for graphical models. In *IJCAI*, pages 224–229.

[Rădulescu et al., 2020] Rădulescu, R., Mannion, P., Roijers, D. M., and Nowé, A. (2020). Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems*, 34(1):1–52.

[Roijers, 2016] Roijers, D. M. (2016). *Multi-Objective Decision-Theoretic Planning*. PhD thesis, University of Amsterdam.

[Roijers et al., 2015a] Roijers, D. M., Whiteson, S., Ihler, A., and Oliehoek, F. A. (2015a). Variational multi-objective coordination. In *NIPS Workshop on Learning, Inference and Control of Multi-Agent Systems*.

[Roijers et al., 2015b] Roijers, D. M., Whiteson, S., and Oliehoek, F. A. (2015b). Computing convex coverage sets for faster multi-objective coordination. *Journal of Artificial Intelligence Research*, 52:399–443.

[Rosenthal, 1977] Rosenthal, A. (1977). Nonserial dynamic programming is optimal. In *Proceedings of the ninth annual ACM symposium on Theory of computing*, pages 98–105.

[Russell and Norvig, 2005] Russell, S. and Norvig, P. (2005). AI a modern approach. *Learning*, 2(3):4.

[Scharpff et al., 2016] Scharpff, J., Roijers, D., Oliehoek, F., Spaan, M., and de Weerdt, M. (2016). Solving transition-independent multi-agent mdps with sparse interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

[Scharpff, 2020] Scharpff, J. C. D. (2020). *Collective Decision Making through Self-regulation: Mechanisms and Algorithms for Self-regulation in Decision-Theoretic Planning*. PhD thesis, Delft University of Technology.

[Verstraeten, 2021] Verstraeten, T. (2021). *A Multi-Agent Reinforcement Learning Approach to Wind Farm Control*. PhD thesis, Vrije Universiteit Brussel.

[Verstraeten et al., 2020] Verstraeten, T., Bargiacchi, E., Libin, P. J., Helsen, J., Roijers, D. M., and Nowé, A. (2020). Multi-agent Thompson sampling for bandit applications with sparse neighbourhood structures. *Scientific reports*, 10(1):1–13.

[Verstraeten et al., 2021] Verstraeten, T., Daems, P.-J., Bargiacchi, E., Roijers, D. M., Libin, P. J., and Helsen, J. (2021). Scalable optimization for wind farm control using coordination graphs. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1362–1370.

[Visser et al., 2014] Visser, A., Ito, N., and Kleiner, A. (2014). Robocup rescue simulation innovation strategy. In *Robot Soccer World Cup*, pages 661–672. Springer.

Active learning for reducing labeling effort in text classification tasks

Pieter Floris Jacobs[1 [0000−0002−8835−6356]],
Gideon Maillette de Buy Wenniger[2,1 [0000−0001−8427−7055]],
Marco Wiering[1 [0000−0003−4331−7537]],
Lambert Schomaker[1 [0000−0003−2351−930X]]

[1] University of Groningen, Groningen, The Netherlands
p.f.jacobs AT student.rug.nl;
{l.r.b.schomaker, m.a.wiering } AT rug.nl
[2] Open University of the Netherlands
gemdbw AT gmail.com

Abstract. Labeling data can be an expensive task as it is usually performed manually by domain experts. This is cumbersome for deep learning, as it is dependent on large labeled datasets. Active learning (AL) is a paradigm that aims to reduce labeling effort by only using the data which the used model deems most informative. Little research has been done on AL in a text classification setting and next to none has involved the more recent, state-of-the-art Natural Language Processing (NLP) models. Here, we present an empirical study that compares different uncertainty-based algorithms with BERT$_{base}$ as the used classifier. We evaluate the algorithms on two NLP classification datasets: Stanford Sentiment Treebank and KvK-Frontpages. Additionally, we explore heuristics that aim to solve presupposed problems of uncertainty-based AL; namely, that it is unscalable and that it is prone to selecting outliers. Furthermore, we explore the influence of the query-pool size on the performance of AL. Whereas it was found that the proposed heuristics for AL did not improve performance of AL; our results show that using uncertainty-based AL with BERT$_{base}$ outperforms random sampling of data. This difference in performance can decrease as the query-pool size gets larger.

Keywords: Active Learning · Text Classification · Deep Learning · BERT.

1 Introduction

Deep Learning (DL) is a field in machine learning in which neural networks with a large number of layers are made to perform complicated human tasks. These networks have to be trained on a large amount of data to be able to learn the underlying distribution of the task they are trying to model. In supervised learning, this data is required to be labeled with the desired output. This allows the network to learn to map the input to the desired output. This study will focus

on an instance of supervised learning, called text classification. Data labeling is usually done manually and can grow to be an expensive and time-consuming task for larger datasets, like those used in DL. This begs the question of whether there is no way to reduce the labeling effort while preserving good performance on the chosen task. Similarly to lossy compression [1], we want to retain a good approximation of the original dataset while at the same time reducing its size as much as possible. More specifically: given a training set, how can we optimally choose a limited number of examples based on the amount of relevant information they contain for the target task?

Conceptually, answering this question requires quantifying the amount of information contained in each data point. This finds its roots, like lossy compression, in information theory [30]. A model trained on limited data has an entropy associated with its target variable predictions. Our goal is to greedily select the data for labeling, while reducing entropy as much as possible, similar to how it is done in research on decision trees [12]. In essence, we aim to incrementally, optimally select a subset of data points; such that the distribution encoded by the learned model maximizes the information gain or equivalently minimizes the Kullback-Leibler divergence [20] with respect to the unknown distribution of the full labeled data. However, there are two problems. First, the labels of the data are not known until labeling, and additional held-out labeled data to aid the selection is typically not available either. This contrasts with the easier case of summarizing a known dataset by a subset of data, in which the Kullback-Leibler divergence of a selected subset with the full set can be measured and minimized. Second, because the parameters of a neural network change during training, predictions and certainty of new data points also change. Because of these two problems, examples can only be greedily selected based on their expected utility for improving the current, incrementally improved model. As the actual labels for examples are lacking before their selection, their real utility cannot be known during selection. Therefore, only proxies for this utility such as model uncertainty can be used, as discussed next.

A machine-learning technique called Active Learning (AL) [29] can be used to combat these problems. In AL, a human labeler is queried for data points that the network finds most informative given its current parameter configuration. The human labeler assigns labels to these queried data points and then the network is retrained on them. This process is repeated until the model shows robust performance, which indicates that the data that was labeled is a sufficient approximation of the complete dataset. There are multiple types of informativeness by which to determine what data to query the oracle for. For instance calculating what results in the largest model change [3] or through treating the model as a multi-arm bandit [2]. However, the existing literature predominantly utilizes different measures of model uncertainty [5,7,8,9,35], which is also done in this research. Bayesian probability theory provides us with the necessary mathematical tools to reason about uncertainty, but for DL has its complications. The reason is that (typical) neural networks, as used for classification and regression, are discriminative models. These produce a single

output, a so called point estimate. Even in the case of softmax outputs this is not a true probability density function [7,8]. Another view on this is that modern neural networks often lack adequate *confidence calibration*, meaning they fail at predicting probability estimates representative of the true correctness likelihood [13].

This poses a problem to Bayesian probability theory as it prevents us from being able to perform Bayesian inference. With Bayesian inference we can determine the probability of a certain output y* given a certain input point x*:

$$p(y*|x*, X, Y) = \int p(y*|x*, \omega)p(\omega, X, Y)d\omega \quad (1)$$

Unfortunately, for the discriminative neural network models there is no probability distribution: the output is always the same for a given input. What is more, even if we suppose the network was generative (Eq. 1), the integral is not analytically solvable due to the fact that we need to integrate over all possible parameter settings ω. However, it can be approximated. Existing literature has explored different methods of achieving this, with Monte Carlo Dropout (MCDO) being the most popular one [5,8,36]. In MCDO, the network applies dropout [33] to make the network generative. Multiple stochastic forward passes are performed to produce multiple outputs for the same input. The outputs can then be used to summarize the uncertainty of the model in a variety of ways.

This research uses the MCDO approximation to compare different uncertainty-related AL query methods for text classification, noting there is still little literature on the usability of AL for modern NLP models. We strive to answer the following research question:

Research Question. *How can uncertainty-based Active Learning be used to reduce labeling effort for text classification tasks?*

Where previous literature focused on comparing AL strategies on small datasets and on the test accuracy of the final classifier, this paper will try and explore the usability of AL on a real-world setting, in which factors like the effect of transfer learning and considerations such as scalability have to be taken into account. The goal is to reach a performance similar to the state-of-the-art text-classification models that use a large randomly sampled set of labeled examples as training set. This should show whether AL can be applied to reduce labeling effort.

2 Related Work

Active Learning applied to Deep Learning for Image Classification Multiple methods of incorporating AL into Deep Neural Networks (DNNs) have been proposed in the past. Most of these focus on image classification tasks.

Houlsby et al. [15] proposed an information theoretic approach to AL: Bayesian Active Learning by Disagreement (BALD). In hopes of achieving state-of-the-art performance and making minimal approximations for achieving

tractability, they used a Gaussian process classifier and compared the performance of BALD to nine other AL algorithms. Their findings included that BALD, which we use in this study, makes the smallest number of approximations across all tested algorithms.

Gal et. al [9] used a Bayesian convolutional network together with MCD to be able to approximate Bayesian inference and thereby proposed an AL framework that makes working with high dimensional data possible. They compared results of a variety of uncertainty-based query functions (including BALD and variation ratio) to random sampling and found that their approach to scaling AL to be able to use high dimensional data was a significant improvement to previous research, with variation ratio achieving the best results.

Drost [5] provided a more extensive discussion of the different ways of incorporating uncertainty into DNNs. He tried to learn which way of computing the uncertainty for DNNs worked best. Using a convolutional neural network, he compared the use of dropout, batch normalization, using an Ensemble of NNs and a novel method named Error Output for approximating Bayesian inference. His main conclusion was that using dropout, batch normalization and ensembles were all useful ways of lowering uncertainty in model predictions. He found that the Ensemble method provided the best uncertainty estimation and accuracy but that it was very slow to train and required a large amount of memory. He concluded MCDO, which is what we use in this study, to be a promising strategy of uncertainty estimation, albeit that one has to take into account slow inference times.

Gikunda and Jouandeau [10] explored an approach for preventing the selection of outlier examples. They combined the uncertainty measure with a correlation measure, measuring the correlation of each unlabeled example with all other unlabeled examples. A higher correlation indicated that an example was less likely to be an outlier. Their method is similar to using a local KNN-based example density as discussed in [39], which is one of the methods we used in this work. The main difference with the KNN-density approach is that their correlation-based density does not consider local neighborhoods in the density estimation. As uncertainty measure they used so-called sampling margin, which is based on the difference in probability between the most likely and second most likely class according to softmax outputs. This is somewhat similar to variation ratio, but does not use stochastic forward passes. It uses plain softmax outputs instead, making it quite distinct from the dropout-sampling based approach we adopt in this work.

Active Learning applied to Deep Learning for Text Classification
A survey of deep learning work on using AL for text classification is given in [28]. They present a taxonomy of different query functions, including those focused on prediction and model uncertainty that we use. They also discuss the incorporation of word embeddings into DNN-based AL, which is something that we attempt in this study.

BERT is used in combination with AL in [6]. They presented a large-scale empirical study on AL techniques for BERT-based classification, covering a diverse set of AL strategies and datasets; focusing on binary text classification with small annotation budgets. They concluded that AL can be used to boost BERT performance.

Active Learning for Regression
Whereas our work is on classification, dropout-based AL can be adapted for regression as well, and this was done by [37]. They used the set of T sample predictions from the forward passes to compute sample standard deviation for the T predictions, using this as a measure of uncertainty. Evaluation was done on standard open multivariate datasets of the UCI Machine Learning repository.

Confidence Calibration
Dropout sampling as used in this work aims to solve the problem that softmax outputs are not reliable representations of the true class probabilities. This problem is known as *confidence calibration*, and dropout sampling is not the only solution to it.

Guo et. al [13] evaluated the performance of various post-processing techniques that took the neural network outputs and transformed them into values closer to representative probabilities. They found that in particular a simplified form of *Platt Scaling*, known as *temperature scaling*, was effective in calibrating predictions on many datasets. This method conceptually puts a logistic regression model with just one learnable 'temperature' parameter behind the softmax outputs, and is trained by optimizing negative log likelihood (NLL) loss over the validation set. It thus learns to spread out or peak the probabilities further in a way that helps to decrease NLL loss, thereby as a side-effect increasing calibration. Recently, using a new procedure inspired by Platt Scaling, Kuleshov et. al [19] generalized an effective approach for confidence calibration to be usable for regression problems as well.

3 Methods

This section will go on to describe the general AL loop, the model architecture, the used query functions, the implemented heuristics, and finally the experimental setup.

3.1 Active Learning

An implementation of the general AL loop/round is shown in Appendix A.2 (Algorithm 1). It consists of four steps:

1. **Train:** The model is reset to its initial parameters. After this, the model is trained on the labeled dataset $\mathcal{L}$. The model is reset before training because otherwise the model would overfit on data from previous rounds [16].

2. **Query:** A predefined query function is used to determine what data is to be labeled in this AL round. As discussed, this can be done in various ways, but the guiding principle is that the data that the model finds most useful for the chosen task gets queried.
3. **Annotate:** The queried data is parsed to a human expert, often referred to as the oracle. The oracle then labels the queried examples.
4. **Append:** The newly-labeled examples are transferred from the unlabeled dataset $\mathcal{U}$ to $\mathcal{L}$. The model is now ready to be retrained to recompute the informativeness of the examples in $\mathcal{U}$ now that the underlying distribution of $\mathcal{L}$ has been altered.

Please note that the datasets used for the experiments (Section 3.5) were fully labeled and the annotation step thus got skipped in this research. $\mathcal{U}$ existed out of labeled data that was only trained on from the moment it got queried. This was done to speed up the process and to enable scalable and replicable experiments with varying experimental setups.

3.2 Model Architecture

BERT The model used to classify the texts was BERT_{base} [4], a state-of-the-art language model which is a variant of the Transformer model [38]. Specifically, we used the uncased version of BERT_{base}, as the information of capitalization and accent markers was judged to be not helpful for the used tasks and datasets. Due to computational constraints, only the first sentence of the used texts was put into the tokenizer and the maximal length to which the tokenizer either padded or cut down this sentence was set to 50. To better deal with unknown words and shorter text, we used the option of the BERT_{base} tokenizer to make use of special tokens for sentence separation, padding, masking and to generalize unknown vocabulary. Finally, a softmax layer was added to the end of BERT_{base}, which is essential as the implemented query functions (Section 3.3) compute uncertainty based on sampled output probability distributions.

Monte Carlo Dropout Monte Carlo dropout (MCDO) is, as discussed in Section 1, a technique that enables reasoning about uncertainty with neural networks. Dropout [33] essentially 'turns off' neurons during the forward pass with a predefined probability. Dropout is normally used during training to prevent overfitting and create a more generalized model. In MCDO though, it is used to approximate Bayesian inference [8] through creating T predictions for all data points, using T slightly different models induced by different dropout samples. The result of these so-called stochastic forward passes (SFP's) can then be used by the query function to compute the uncertainty, as will be explained in Section 3.3. The way MCDO is incorporated in the AL loop is shown in green in the Appendix (Algorithm 2). BERT_{base} has two different types of dropout layers: hidden dropout and attention dropout. Both were turned on when performing a stochastic forward pass. Note that there are other ways of approximating Bayesian inference with neural networks. Frequently used ones are:

– Having an ensemble of neural networks vote on the label [18].
– Monte Carlo Batch Normalization (MCBN) [35].

MCDO was chosen over the ensemble method due to it being easier to implement and quicker to train. MCBN was not chosen as it has been shown to be more inconsistent than MCDO [5].

Sentence-BERT Textual data offers the advantage of having access to the use of pre-trained word embeddings. These are learned representations of words into a vector space in which semantically similar words are close together. Textual embeddings can be computed in a variety of ways. BERT specific ones include averaging the pooled BERT embeddings and looking at the BERT CLS token output. Other more general ways are averaging over Glove word embeddings [25] and averaging embeddings created by a Word2Vec model [22]. We have opted to make use of Sentence-BERT [26], a Siamese BERT architecture trained to produce embeddings that can be adequately compared using cosine-similarity. For our purposes this provides better performance than the other embedding computations. Sentence-BERT was used separately from the previously discussed BERT_{base} model, and was used only for assigning embeddings to each sentence in the dataset that were used by the heuristics described in Section 3.4.

3.3 Query Functions

The query functions determine data selection choices of the model in the AL loop. This paper will focus on functions that reason about uncertainty, obtained from approximated Bayesian distributions [8]. For every data point, the distribution is derived from T stochastic forward passes and resulting T (in our case) softmax probability distributions. The following subsections will go on to discuss the implemented query functions. One is encouraged to look at [7] for an extensive discussion that highlights the difference between these functions.

Variation Ratio The variation ratio is a measure of dispersion around the class that the model predicts most often (the mode). The intuition here is that the model is uncertain about a data point when it has predicted the mode class a relatively small number of times. This indicates that it has predicted other classes a relatively large number of times. Equation 2 shows how the variation ratio is computed, where f_x denotes the mode count and T the number of stochastic forward passes.

$$v[x] = 1 - \frac{f_x}{T} \tag{2}$$

The function attains its maximum value when the model predicts all classes an equal amount of times and its minimum value when the model only predicts one class across all stochastic forward passes. Variation ratio only captures the uncertainty contained in the predictions, not the model, as it only takes into account the spread around the most predicted class. It is thus a form of predictive uncertainty.

Predictive Entropy Entropy $H(x)$ in the context of information theory is defined as:

$$H(x) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{3}$$

This formula expresses the entropy in bits per symbol to be communicated, in which $p(x_i)$ gives the probability of the i-th possible value for the symbol. Entropy is used to quantify the information of data. In our case we want to know the chance of the model classifying a data point as a certain class given the input and model parameters ($p(y = c|\mathbf{x}, \boldsymbol{\omega})$). We can compute this chance by averaging over the softmax probability distributions across the T stochastic forward passes. This adjusted version of entropy is denoted in Equation 4, where $\hat{\omega}_t$ denotes the stochastic forward pass t, and c the number associated to the class-label.

$$\begin{aligned} H[y|\mathbf{x}, \mathcal{D}_{train}] = -\sum_{c} \left(\frac{1}{T} \sum_{t} p(y = c|\mathbf{x}, \hat{\boldsymbol{\omega}}_{\boldsymbol{t}}) \right) \\ \log \left(\frac{1}{T} \sum_{t} p(y = c|\mathbf{x}, \hat{\boldsymbol{\omega}}_{\boldsymbol{t}}) \right) \end{aligned} \tag{4}$$

To exemplify: in binary classification, the predictive entropy is highest when the model its softmax classifications consist of T times [0.5, 0.5]. In that case, expected surprise when we would come to know the real class-label is at its highest. The uncertainty is computed by averaging over all predictions and thus falls under predictive uncertainty.

Bayesian Active Learning by Disagreement Predictive entropy (Section 3.3) is used to quantify the information in one variable. Mutual information or joint entropy is very similar but is used to calculate the amount of information one variable conveys about another. In our case, we'll be looking at what the average model prediction will convey about the model posterior, given the training data. This is a form of conditional mutual information, the condition or the third variable being the training data $\mathcal{D}_{train}$. Houlsby et al. [15] used this form of mutual information in an AL setting and dubbed it Bayesian active learning by disagreement (BALD).

$$\begin{aligned} I[y, \omega|\mathbf{x}, \mathcal{D}_{train}] = -\sum_{c} \left(\frac{1}{T} \sum_{t} p(y = c|\mathbf{x}, \hat{\boldsymbol{\omega}}_{\boldsymbol{t}}) \right) \\ \log \left(\frac{1}{T} \sum_{t} p(y = c|\mathbf{x}, \hat{\boldsymbol{\omega}}_{\boldsymbol{t}}) \right) \\ -\frac{1}{T} \sum_{c,t} p(y = c|\mathbf{x}, \hat{\boldsymbol{\omega}}_{\boldsymbol{t}}) \\ \log p(y = c|\mathbf{x}, \hat{\boldsymbol{\omega}}_{\boldsymbol{t}}) \end{aligned} \tag{5}$$

The difference between Equations 5 and 4 is that the conditional entropy is subtracted from the predictive entropy. The conditional entropy is the probability of the full output being generated from the training data and the input. This is the reason we do not average the predictions for every single class. We first sum over all classes, so that we do not average over the model parameters for every single class and thus take into account the fact that we are looking at the chance of the complete probability distribution being generated.

BALD is maximized when the T predictions are strongly disagreeing about what label to assign to the example. So in the binary case, it would be highest when the predictions would alter between [1,0] and [0,1] as these two predictions are each others complete opposite. Unlike the variation ratio and predictive entropy, BALD is a form of model uncertainty. When the softmax outputs would be equal to T times [0.5,0.5], the minimal BALD value would be returned as the predictions are the same and the model is thus very confident about its prediction.

3.4 Heuristics

Redundancy Elimination In AL, a larger query-pool size (from now on referred to as q) results in the model being retrained less and the uncertainties of examples being re-evaluated less frequently. Consequently, the model gets to make less informed decisions as it uses less up-to-date uncertainty estimates. Larger q could therefore theoretically cause the model to collect many similar examples for specific example types with high model uncertainty in an AL round. Say for instance we were dealing with texts about different movie genres. Suppose the data contained a lot of texts about the exact same movie. When the model would be uncertain about this type of text, a large q would result in a large amount of these texts getting queried. This could be wasteful, as querying this type of text a small amount of times would likely result in the model no longer being uncertain about that type of text. Note however, that low model uncertainty by itself is no guarantee for robustly making accurate predictions for a type of examples. Yet provided such robust performance is achieved, additional examples of the same type would be a waste.

The above could form a problem as although a smaller q should theoretically provide us with better results, it also requires more frequent uncertainties re-computation. Every computation of the uncertainties requires T stochastic forward passes on the unlabeled dataset $\mathcal{U}$. This entails that, next to the computation, the time required to label a dataset would increase as well, which is not in line with our goal. In hopes of improving performance with larger q, we propose two heuristics:

1. Redundancy Elimination by Training (RET)
2. Redundancy Elimination by Cosine Similarity (RECS)

For both of these heuristics, a new pool, which we will refer to as the redundancy-pool $\mathcal{RP}$, is introduced. The query-pool $\mathcal{QP}$ will be a subset of $\mathcal{RP}$ of which we will try to select the most dissimilar examples.

RET tries to eliminate redundant data out of $\mathcal{RP}$ by using it as a pool to retrain on. The data point with the highest uncertainty is trained on for one epoch and then the uncertainties of the examples in $\mathcal{RP}$ are recomputed. This process gets repeated until $\mathcal{QP}$ is of the desired size. Note that although this strategy seems similar to having a q of one, it is less computationally expensive as only the uncertainties for the examples in $\mathcal{RP}$ have to be recomputed (which also shrinks after each repetition). Algorithm 3 of Appendix A.2 shows how RET is integrated in the AL loop.

The main purpose of RET is to enable the use of larger q. However, one needs to be mindful of the fact that when q is increased, $\mathcal{RP}$ is to be increased in size well. This being due to the fact that smaller differences between the sizes of $\mathcal{RP}$ and $\mathcal{QP}$ result in less influence of the heuristic. In the RET algorithm, forward passes over $\mathcal{RP}$ contribute to the total amount of forward passes. Furthermore, this contribution increases linearly with the redundancy-pool size ($|\mathcal{RP}|$) and in practice coupled query-pool size q. Using $|\mathcal{RP}| = 1.5 \times q$, this contribution starts to dominate the total amount of forward passes (approximately) once $q > \sqrt{|\text{data}|}$. This is explained in more detail in Appendix A.1. This limits its use for decreasing computation by increasing q. Because of this, RECS is aimed at being computationally cheaper.

Instead of retraining the model and constantly taking into account recomputed uncertainties, RECS makes use of the sentence embeddings created by Sentence-BERT (Section 3.2). The assumption made is that semantically similar data conveys the same type of information to the model. The examples are selected based on their cosine similarity to other examples. $\mathcal{RP}$ is looped through and examples are only added to $\mathcal{QP}$ if their cosine similarity to all other points that are already in $\mathcal{QP}$ is lower than the chosen threshold l. If not enough examples are selected to get the desired q, the threshold gets decreased by 0.01. Algorithm 4 of Appendix A.2 shows how this heuristic is added to the AL loop.

Sampling by Uncertainty and Density (SUD) Schomaker and Oosten [24] showed that the distinction between separability and prototypicality is important to account for. In their use case of the SVM, data points that had a high margin to the decision boundary were not always representative of the class prototype. Uncertainty sampling also tries to sample examples close to the decision boundary, but has been shown to often select outliers [27,34]. Outliers contain a lot of information that the model has not encountered yet, but this information is not necessarily useful. As with the previously described RECS heuristic, we hypothesize that semantically similar sentences provide the same type of information. In that situation, outliers are very far from other examples in embedding space.

Zhu et. al [39] proposed a K-Nearest-Neighbor-based density approach called Sampling by Uncertainty and Density (SUD) to avoid outliers based on their distance in embedding space. In this approach, the mean cosine similarity between every data point and its K most similar neighbors is computed. A low value indicates that a data point is not very similar to others. This value

is then multiplied with the uncertainty and the dataset is sorted based on this Uncertainty-Density measure. They showed that this measure improved performance of the maximum entropy model classifier. We will explore whether this approach also works for BERT combined with the embeddings computed by Sentence-BERT. The adjusted pseudocode is shown in Appendix A.2 (Algorithm 5).

3.5 Experimental Setup[3]

Data Two datasets were used to validate and compare the performance of the different AL implementations. Table 1 shows an overview of the amount of examples and classes of each dataset. The first of the used datasets was the

Table 1. An overview of the two datasets used in the experiments

Dataset	Examples	Number of Classes
SST	11,850	5
KvK	2212	15

Stanford Sentiment Treebank [32] (SST). SST exists out of 215,154 phrases from movies with fine-grained sentiment labels in the range of 0 to 1. These phrases are contained in the parse trees of 11,855 sentences. Only these full sentences were used in the experiments, and the sentiment labels were mapped to five categories in the following way:

- $0 \leq \text{label} < 0.2$: very negative
- $0.2 \leq \text{label} < 0.4$: negative
- $0.4 \leq \text{label} \leq 0.6$: neutral
- $0.6 < \text{label} \leq 0.8$: positive
- $0.8 < \text{label} \leq 1$: very positive

Use of the SST dataset was motivated by its size as well as by it being a benchmark for language models. It allowed for the evaluation of AL for a larger dataset and for comparison with results found in related work such as [23]. This helped to check whether BERT_{base} was achieving desirable performance.

The second dataset that was used consists of the descriptions of companies located in Utrecht. The companies are all registered at the Dutch Chamber of Commerce, or Kamer van Koophandel (KvK) and were mapped to their corresponding SBI-code. The SBI code denotes the sector a company operates in, as defined by the KvK. The HTML of the companies websites was scraped and the meta content that was tagged as the description was extracted. In nearly all cases, this contained a short description about what the company was involved in. Note that only English descriptions were used. The KvK dataset provided

[3]The code used for the experiments can be found at `https://github.com/Pieter-Jacobs/bachelor-thesis`

us with the opportunity to evaluate AL for a classification problem with a large amount of classes as well as the ability to compare results between a dataset with a limited number of examples and one with a relatively large amount of examples (SST). Testing AL on a dataset with a limited number of examples was deemed necessary due to the fact that most of the positive results found in related work were achieved by making use of very small datasets. The dataset will not be shared and is not available online due to the fact that it was constructed as part of an internship at Dialogic.

Evaluation Metrics To evaluate and compare the performance of the different AL strategies, two evaluation metrics were reported: the accuracy and an altered version of the deficiency metric proposed in [39].

The variant of deficiency that was used is shown in Eq. 6, in which n denotes the amount of accuracy scores, $acc(R)$ denotes the accuracy of the reference strategy and $acc(C)$ the accuracy of the strategy to be compared to this reference strategy. In our case, n is equal to $\frac{|\mathcal{U}|}{q} + 1$ ($+1$ comes from the accuracy achieved after training on the seed), as we computed the test accuracy after every AL round.[4] Furthermore, instead of using the accuracy that was achieved in the final AL round for $acc(C)$ and $acc(R)$ like [39], we use the overall maximum accuracy. This accounts for the fact that the last achieved accuracy in a classification task is not necessarily the best value, while still returning a metric which provides a summary of the entire learning curve. This in turn means that a decrease/increase in its value is analogical to a decrease/increase in overall performance of the comparison strategy. However, the deficiency does not convey whether there were points at which the accuracy of a strategy was higher than usual and would serve as a good point to cut-down the dataset to reduce labeling effort. A deficiency of <1 indicates a better performance than the reference strategy whereas a value of >1 indicates a worse performance.

$$DEF(AL, R) = \frac{\sum_{t=1}^{n}(max(acc(R)) - acc_t(C)}{\sum_{t=1}^{n}(max(acc(R)) - acc_t(R))} \tag{6}$$

Experiments The goal of the experiments was to answer the question of whether overall labeling effort could be reduced through making use of AL. We split this into the following three sub-questions:

1. Does AL achieve better performance with less data when compared to plain random sampling?
2. What is the relation between query-pool size q and the achieved performance?
3. Do the proposed heuristics (SUD, RET, RECS) improve the performance of AL?

[4] For our experiments, this resulted in our n ranging from 20 to 191 for the SST dataset and from 17 to 152 for the KvK dataset (the used q can be found in Section 3.5).

Table 2. The statistical setup used for both datasets. The percentages used are relative to the full dataset size.

Dataset	Seed	$\mathcal{U}$	Dev	Test
SST	594 (5%)	7951 (67%)	1101 (9%)	2210 (19%)
KvK	111 (5%)	1659 (75%)	221 (10%)	221 (10%)

The statistical setup used for the experiments can be found in Table 2. The setup for SST was based on the proposed setup in [32]. To reiterate, the following AL strategies were implemented:

1. Variation Ratio (Section 3.3)
2. Predictive Entropy (Section 3.3)
3. BALD (Section 3.3)
4. RET (Section 3.4)
5. RECS (Section 3.4)
6. SUD (Section 3.4)

To answer subquestion 1, these strategies were compared to the performance of random sampling using a q of 1% of the dataset size. For subquestion 2, the three query functions were be compared across three q: 0.5%, 1% and 5% of the dataset size. Finally, to be able to answer subquestion 3, RET, RECT and SUD were compared with a q of 1%. As RET, RECS and SUD were meant as additions to general problems of uncertainty-based AL, they were only tested for the variation ratio query function. This function was chosen, because it was reported in [7] to give the best result. To make the results more generalizable, all the experiments mentioned above were run three times.

Moreover, to test the assumption of the RECT strategy, we measured whether there was a relation between how the model softmax predictions changed towards the one-hot vector of the actual label and the cosine similarity to the data point that was trained on. The relationship was quantified by means of Kendall's τ between the ranking of the examples based on which one had the largest change in KL divergence after training on the top example and the ranking of the examples based on cosine similarity to the example being trained on.

Hyperparameters Table 4 gives an overview of used hyperparameters. Model weights were randomly initialized using the various PyTorch initialization defaults for the respective model components. In addition to the randomness of weight initialization, randomness determines dropout choices during training. These two forms of randomness influence model performance. For each system/setting, we averaged results over three repeated runs which were identical except for these random elements. This helps to prevent false conclusions due to performance differences caused by effects of these elements.

Both dropout rate and l (the cosine-similarity threshold used in RECS) were chosen based on a grid search across both datasets. The amount of stochastic forward passes T was based on [6] and was set to 10 across all experiments.[5] Early

[5]Larger values up to 100 were tested, but induced much larger training times without noteworthy performance gains.

Table 3. The amount of epochs used for early stopping for the different datasets.

Dataset	# Epochs
SST	15
KvK	25

Table 4. Hyperparameters values

Parameter	Value
Dropout rate	0.2
T	10
l	0
β_1, β_2	0.9, 0.999
ϵ	$1 * 10^{-8}$
Learning rate	$2 * 10^{-5}$
Batch size	128
$\mathcal{RP}$ size	$1.5*q$
Embedding dim.	768

stopping was applied on each training phase of the AL loop, Table 3 shows the amount of epochs used for each dataset. The model yielding the lowest validation loss across all epochs was used for evaluation and uncertainties computation. Note that in a normal AL setting, validation sets are usually not available due to the labelling effort required and this strategy would be less feasible.

The Adam algorithm [17] was used for optimization and its learning rate was tuned based on the CLR method [31]. The best performing computationally feasible batch size (128), out of the tried batch sizes (32, 64, 128, 256), was used in all experiments. The betas and ϵ were set to their default values. The size of $\mathcal{RP}$ was chosen arbitrarily, determining its optimal choice is left future research.

Finally, dimensionality reduction using PCA was tried to determine whether this would result in better class-separability. For every data point in the full dataset, the classes of the group of ten most similar data points (based on cosine similarity) were determined. By maximizing the average of the number of within-group same-class data points, the used dimensionality was determined.

4 Results

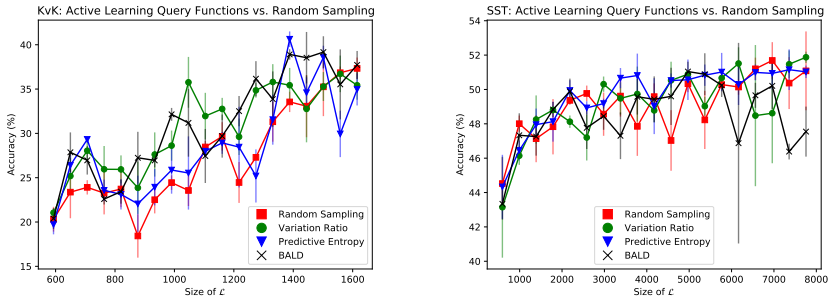
This section will go onto visualize and describe the achieved results for all three experiments described in Section 3.5. Note that for all figures, the results were averaged over three runs with the error bars showing one standard deviation. Furthermore, all deficiencies were rounded to two decimal places. For deficiency values <1 (improvements over the reference strategy), we show the smallest value in the comparison in bold. For the sake of readability and to keep graph points aligned, in the graphs for query-pool sizes of 0.5% and 1% the points shown are respectively those at every 10th and 5th and interval.

4.1 Active Learning

Figure 1a shows how the query functions performed on the KvK dataset. All query functions outperform random sampling when the labeled dataset is less

than 200 examples large. After this, in particular BALD and variation ratio continue to mostly outperform random sampling until near the maximum labeled data size. Notably, many of the performance differences are larger than one standard deviation.

Figure 1b shows how random sampling and the implemented query functions performed on the SST dataset. On this dataset the results for the random sampling baseline and the other systems is much smaller, and there does not seem to be a clear winner.



(a) (b)

Fig. 1. The achieved test accuracy on the KvK dataset (a) and on the SST dataset (b) by random sampling and the uncertainty-based query functions.

Finally, the deficiencies shown in Table 5 show a positive result (< 1) for all query functions except for predictive entropy for the SST dataset. Matching the graphs, the performance gains as measured by the deficiency scores are overall more substantial on the KvK dataset. BALD has the lowest deficiency for both datasets.

Table 5. The deficiencies (Eq. 6) of the uncertainty-based query functions. Random sampling was the reference strategy.

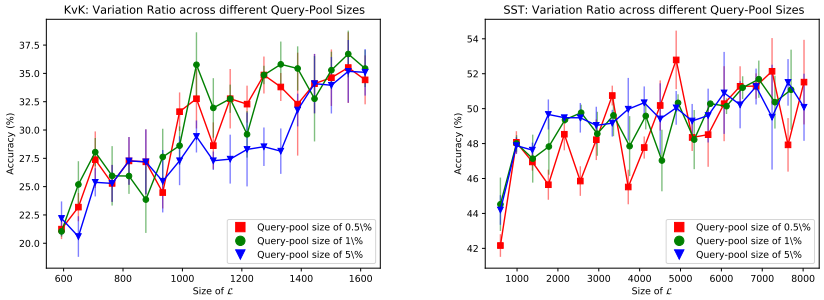
Dataset	VR	PE	BALD
SST	0.95	1.01	**0.89**
KvK	0.67	0.9	**0.64**

4.2 Query-pool Size

Figure 2a shows the performance of variation ratio across different q when used on the KvK dataset. In the middle range of the graph, variation ratio with a q

of 5% has a worse performance than the other q. The q of 0.5% and 1% achieve similar performance with the accuracy scores always staying within one standard deviation of each other.

Figure 2b shows the performance of the different q on the SST dataset. The performance of variation ratio with a q of 0.5% fluctuates more when compared to the other q. Moreover, it results in an overall worse performance when compared to the other sizes. The q of 5% shows to have the best and most consistent performance over the whole learning curve in terms accuracy. However, the q of 0.5% manages to outperform the other q at about 5000 labeled examples.



(a) (b)

Fig. 2. The achieved test accuracy on the KvK dataset (a) and the SST dataset (b) by using the variation ratio query function with different q.

The deficiencies for the different q across both datasets are shown in Table 6. For the SST dataset, the q of 5% had a lower deficiency across the learning curve whereas the q of 0.5% shows a relatively high deficiency. For the KvK dataset however, we see that the q of 5% has a relatively high deficiency when compared to the similarly performing q of 0.5% and 1%.

Table 6. The achieved deficiencies (Eq. 6) by the different q for the different datasets. A q of 1% was the reference strategy.

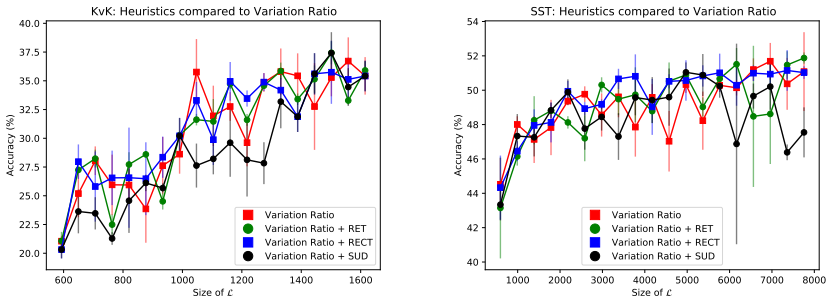
Dataset	0.5%	5%
SST	1.65	**0.62**
KvK	**0.91**	1.33

4.3 Heuristics

Figure 3a shows the performance of using variation ratio with heuristics together with the performance of solely using variation ratio on the KvK dataset (also

shown in Figure 1b). Both RET and RECT show no clear improvement over solely using variation ratio. The same can be gathered from the results of the SST dataset shown in Figure 3b as their accuracy scores stay within one standard deviation for the entire learning curve. Moreover, Table 7 shows that the average Kendall's τ is around 0 with a relatively large standard deviation; indicating that there is no relationship between the compared rankings.

Lastly, SUD shows an overall worse performance for both the SST and KvK datasets. The deficiencies shown in Table 8 also show high values for SUD across both datasets.



(a) (b)

Fig. 3. The achieved test accuracy on the KvK dataset (a) and on the SST dataset (b) by the different heuristics.

Table 7. The mean and the 1 SD range of Kendall's τ from the described ranking experiment across the two datasets (rounded to two decimal places).

Dataset	Mean	σ
SST	0.14	0.33
KvK	0.02	0.47

Table 8. The achieved deficiencies by the different heuristics. Variation ratio was the reference strategy.

Dataset	RET	RECT	SUD
SST	1.02	1.05	1.23
KvK	0.98	**0.96**	1.33

5 Discussion

This research investigated whether AL could be used to reduce labeling effort while at the same time maintaining similar performance to a model trained on a full dataset. To achieve this, the performance and scalability of different AL query-strategies was tested for the state-of-the-art NLP model: BERT.

Conclusions The results showed that uncertainty-based AL can provide improved performance over random sampling for cut-down datasets. This difference was not consistent throughout the whole training curve: at specific points AL outperformed random sampling and at others at it achieved similar performance. BALD was the query function with the overall best performance. This could be the case due to the fact that it is the only query function used which measures model uncertainty. The found results differs from what was found in [7,9], where variation ratio achieved the best overall performance.

Unfortunately, the results found for the KvK dataset show that the found improvement can diminish as query-pool sizes get larger, which corresponds to what was theorized hypothesized in Section 3.4.

Moreover, the two proposed heuristics aimed at improving scalability did not help in improving performance for either dataset and the heuristic aimed at avoiding outliers even resulted in worse performance. This was surprising due to the favorable results found in [39], albeit that they only tested it for training sets of up to 150 examples.

An unexpected result was found in that the assumption that semantically similar data conveyed the same type of information did not hold according to the conducted ranking experiment. A possible explanation for this could be that the texts were not mapped to embeddings in a way in which semantically similar data was close enough to each other. Another curious finding was that for the SST dataset, the smallest q resulted in the worse performance, especially at the beginning of the learning curve. This is counter-intuitive due to the fact that performance seems to suffer from more frequent uncertainty estimates. A potential justification for this could be that updating too frequently at the beginning of the learning curve results in the model not being able to train enough on high frequency classes. This could result in the model focusing too much on the long tail of the class distribution due to the fact that it is more uncertain about texts with low frequency classes at the start of the learning curve. Further research is needed to build a better understanding of this. Conversely, given that AL was shown to have little influence on the achieved accuracy and that most of the differences between the different q are within one standard deviation, one could argue that that the size of q did have an influence on the results whatsoever and that we thus cannot conclude anything from the found results.

From the above, we conclude that uncertainty-based AL with BERT_{base} can be used to decrease labeling effort. This supports what was concluded by [11].

When looking at the bigger picture, we showed that AL can still provide an improvement in performance over random sampling for large datasets. The improvement of performance of AL with BERT is however limited when compared to what it achieved for older NLP models [39,34,27] and even more so when compared to image classifiers [15,5,9]. Performance did show to increase more when used on the KvK dataset. A possible explanation for this is its smaller size. BERT is pretrained on a large amount of data and only needs fine-tuning for achieving good performance on a specific task. Transfer learning models [14] like

BERT have the ability to perform well on new tasks with just a limited amount of data. The power of this few-shot learning also became apparent on a dataset which we decided not to use. Here, BERT was able to get a low validation error on the seed alone, while at the same time having a training accuracy of 100%.

An additional explanation can be found in the nature of the two tasks and their examples. The SST dataset belongs to a sentiment analysis task, with sentiment scores in the range 0–1. These were binned into spans of 0.2 to get a five-class classification task. Furthermore, bag-of-words (BOW) models such as Naive Bayes were shown to perform relatively really well on this task, because specific individual words provide substantial information about the class. As a consequence, each example is actually *compound*: it indirectly provides information about not just that example but about the sentiment contributions of all the words in that example as well. In contrast, the KvK dataset provides is a real classification task as opposed to a regression task converted to classification task, with 15 distinct classes. A subset of words in each example can be expected to be informative for the class label, as opposed to words giving nearly independent contributions as is the case in sentiment analysis.

A limitation of the research was that, due to computational constraints, only the first sentence of texts was used. There were data points where the first sentence did not contain any clear indication of its label. Take for example the following description from the KvK dataset:

*"**Hi, I'm Barbara Goudsmit.** Welcome to my woven world! I am a passionate hand weaver from the Netherlands who loves creating patterns and bringing them to live on my 8-shaft loom."*

This type of data could have resulted in the network learning suboptimal mappings, which could in turn have had an influence on the performance of AL.

Future Research This work focused on classification tasks. A future direction could be to investigate the influence of AL on BERT's performance in the context of regression tasks and also to examine how the proposed heuristics perform there. Moreover, more recent BERT variants, like for instance RoBERTa [21], could be tested to see whether AL still outperforms the random sampling benchmark. Furthermore, the used query functions were mostly developed for and used in computer vision. Query functions aimed at text classification or at the fact that BERT is a pretrained model could be further investigated. Lastly, an important direction for future work remains making AL more scalable by finding ways to preserve performance with larger query-pool sizes.

Acknowledgments

We would like to express our thanks and gratitude to the people at Dialogic (Utrecht) of which Nick Jelicic in particular, for the useful advice on the writing style of the paper and the suggested improvements for the source code.

References

1. Ahmed, W., Natarajan, T., Rao, K.R.: Discrete Cosine Transform. IEEE Transactions on Computers **23**(1), 90–93 (1974)
2. Bouneffouf, D., Laroche, R., Urvoy, T., Féraud, R., Allesiardo, R.: Contextual Bandit for Active Learning: Active Thompson Sampling. In: Proceedings of the 21st International Conference on Neural Information Processing (ICONIP). pp. 405–412 (2014)
3. Cai, W., Zhang, Y., Zhou, J.: Maximizing Expected Model Change for Active Learning in Regression. In: Proceedings - IEEE International Conference on Data Mining, ICDM. pp. 51–60 (2013)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: North American Association for Computational Linguistics (NAACL). pp. 4171–4186 (2019)
5. Drost, F.: Uncertainty Estimation in Deep Neural Networks for Image Classification. Master's thesis, University of Groningen (2020)
6. Ein-Dor, L., Halfon, A., Gera, A., Shnarch, E., Dankin, L., Choshen, L., Danilevsky, M., Aharonov, R., Katz, Y., Slonim, N.: Active Learning for BERT: An Empirical Study. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 7949–7962 (2020)
7. Gal, Y.: Uncertainty in Deep Learning. Master's thesis, University of Cambridge (2016)
8. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In: Proceedings of The 33rd International Conference on Machine Learning. vol. 48, pp. 1050–1059. PMLR (2016)
9. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 1183–1192. PMLR (2017)
10. Gikunda, P.K., Jouandeau, N.: Budget active learning for deep networks. In: Intelligent Systems and Applications. pp. 488–504 (2021)
11. Grießhaber, D., Maucher, J., Vu, N.T.: Fine-tuning BERT for Low-Resource Natural Language Understanding via Active Learning. CoRR **abs/2012.02462** (2020)
12. Gulati, P., Sharma, A., Gupta, M.: Theoretical Study of Decision Tree Algorithms to Identify Pivotal Factors for Performance Improvement: A Review. International Journal of Computer Applications **141**(14), 19–25 (2016)
13. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning. pp. 1321–1330 (2017)
14. Gupta, A., Thadani, K., O'Hare, N.: Effective Few-Shot Classification with Transfer Learning. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 1061–1066 (2020)
15. Houlsby, N., Huszár, F., Ghahramani, Z., Lengyel, M.: Bayesian active learning for classification and preference learning (2011)
16. Hu, P., Lipton, Z.C., Anandkumar, A., Ramanan, D.: Active Learning with Partial Feedback. CoRR **abs/1802.07427** (2018)
17. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. CoRR **abs/1412.6980** (2015)
18. Krogh, A., Vedelsby, J.: Neural Network Ensembles, Cross Validation and Active Learning. In: Proceedings of the 7th International Conference on Neural Information Processing Systems. pp. 231–238. MIT Press (1994)

19. Kuleshov, V., Fenner, N., Ermon, S.: Accurate uncertainties for deep learning using calibrated regression. In: Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 2796–2804 (2018)
20. Kullback, S., Leibler, R.A.: On Information and Sufficiency. The Annals of Mathematical Statistics **22**(1), 79–86 (1951)
21. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR **abs/1907.11692** (2019)
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: Proceedings of Workshop at ICLR. pp. 1–12 (2013)
23. Munikar, M., Shakya, S., Shrestha, A.: Fine-grained sentiment classification using bert (2019)
24. Oosten, J.P., Schomaker, L.: Separability versus prototypicality in handwritten word-image retrieval. Pattern Recognition **47**(3), 1031–1038 (2014)
25. Pennington, J., Socher, R., Manning, C.: GloVe: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014)
26. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. CoRR **abs/1908.10084** (2019)
27. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 441—448 (2001)
28. Schröder, C., Niekler, A.: A survey of active learning for text classification using deep neural networks. CoRR **abs/2008.07267** (2020), `https://arxiv.org/abs/2008.07267`
29. Settles, B.: Active Learning Literature Survey. Synthesis Lectures on Artificial Intelligence and Machine Learning **6**(1), 1–114 (2012)
30. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal **27**(3), 379–423 (1948)
31. Smith, L.N.: No More Pesky Learning Rate Guessing Games. CoRR **abs/1506.01186** (2015)
32. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1631–1642 (2013)
33. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research **15**(56), 1929–1958 (2014)
34. Tang, M., Luo, X., Roukos, S.: Active learning for statistical natural language parsing. In: Proceedings of ACL 2002. pp. 120–127 (2002)
35. Teye, M., Azizpour, H., Smith, K.: Bayesian Uncertainty Estimation for Batch Normalized Deep Networks. In: Proceedings of the 35th International Conference on Machine Learning. vol. 80, pp. 4907–4916. PMLR (2018)
36. Tsymbalov, E., Panov, M., Shapeev, A.: Dropout-Based Active Learning for Regression. Analysis of Images, Social Networks and Texts pp. 247–258 (2018)
37. Tsymbalov, E., Panov, M., Shapeev, A.: Dropout-based active learning for regression. In: Analysis of Images, Social Networks and Texts. pp. 247–258. Springer International Publishing, Cham (2018)

38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 6000—-6010 (2017)
39. Zhu, J., Wang, H., Yao, T., Tsou, B.K.: Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008). pp. 1137–1144 (2008)

Appendix

A.1 RET Algorithm Computational Cost Analysis

The number of forward passes required by the RET algorithm depends on two factors:

1. *Basic passes*: The forward passes required by the "normal" computation of uncertainty at the beginning of the computation for every query-pool.
2. *RP passes*: The forward passed required for intermediate updates, using the redundancy pool RP.

In this analysis we will assume that the size of the redundancy pool $|\mathcal{RP}|$ is chosen as a factor $f > 1$ of the size of the query-pool q. A reasonable assumption, considering that making $|\mathcal{RP}|$ larger than needed incurs unnecessary computational cost, whereas a too small value is expected to diminish the effect of the RET algorithm. We furthermore notice that given this assumption, and assuming a fixed total number of examples to label, there are two factors influencing the required amount of *RP passes*:

- Linearly increasing the query-pool size and coupled redundancy pool size causes a quadratic increase in the number of required forward passes per query pool round.
- At the same time, a linearly increased query-pool size also induces a corresponding linear decrease in the number of required query-pool rounds.

We will see that these two factors will cause a net linear contribution to the number of *RP passes* starts causing a net increase of total passes once the query-size comes above a certain value. Looking at (1) more precisely, the amount of passes over $\mathcal{RP}$ that needs to be performed per query-pool round can be computed as an *arithmetic progression*:

$$|\mathcal{RP}| + (|\mathcal{RP}| - 1) + (|\mathcal{RP}| - 2) + \ldots + (|\mathcal{RP} - q) = \tag{7}$$

$$\frac{1}{2} \times (q+1) \times (|\mathcal{RP}| + |\mathcal{RP}| - q) = \tag{8}$$

$$\frac{1}{2} \times (q+1) \times ((2f-1) \times q) = \tag{9}$$

$$\frac{1}{2} \times (q+1) \times f' \times q) = \tag{10}$$

$$\frac{1}{2} \times f' \times (q^2 + q)) \tag{11}$$

Let's assume we use $f = 1.5$ (as also used in our experiments), and consequently, $f' = 2f - 1 = 2$. The number of forward passes over $\mathcal{RP}$ then becomes exactly $q^2 + q$.

The complexity can then be expressed by the following formula:

$$T \times \lceil \frac{\#\text{Samples}}{q} \rceil \times (|\text{data}| + q^2 + q) \tag{12}$$

This can be approximately rewritten as:

$$T \times \#\text{Samples} \times (\frac{|\text{data}|}{q} + \frac{q^2 + q}{\text{query-pool}}) = \tag{13}$$

$$T \times \#\text{Samples} \times (\frac{|\text{data}|}{q} + q + 1) \tag{14}$$

Note that the second term query-pool-size + 1 only starts dominating the number of forward passes in this formula as soon as:

$$q + 1 \approx q > \frac{|\text{data}|}{q}$$

This is the case when

$$q > \sqrt{(}|\text{data}|)$$

Until then, the computational gains of less *basic passes* outweighs the cost of more *RP passes*. In practice though, this may happen fairly quickly. For example, assuming we have a data size of 10000 examples, and we use as mentioned $q = 1.5| \times \mathcal{RP}|$, then as soon as $q \geq 100$ the increased computation of the *RP passes* starts dominating the gains made by less *basic passes* when further increasing the query-pool size, and the net effect is that the total amount of computation increases.

In summary, for the RET algorithm, *RP passes* contribute to the total amount of forward passes. Furthermore, this contribution increases linearly with redundancy-pool size and coupled query-pool size, and starts to dominate the total amount of forward passes once redundancy-pool-size $> \sqrt{\text{data-size}}$. This limits its use for decreasing computation by increasing the query-pool size.

A.2 Algorithms

Algorithm 1 The general AL loop.

Input Labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$, the unlabeled data $\mathcal{U} = \{(x_i, \emptyset)\}_i^n$ and the untrained classifier $f(x; \theta)$.
Output Fully labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$ and trained classifier $f(x; \theta)$

$n \leftarrow$ Desired length of $\mathcal{L}$
$q \leftarrow$ Query-pool size
$Q(x) \leftarrow$ Query Function
while $\mathcal{L}$ length $< n$ **do**
    Retrain $f(x; \theta)$ on $\mathcal{L}$
    Sort $\mathcal{U}$ based on $Q(\mathcal{U})$
    Let Oracle assign labels to $\mathcal{U}_0^q$
    Insert $\mathcal{U}_0^q$ into $\mathcal{L}$
    Remove $\mathcal{U}_0^q$ from $\mathcal{U}$
end while

Algorithm 2 The AL loop with MCD.

Input Labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$, the unlabeled data $\mathcal{U} = \{(x_i, \emptyset)\}_i^n$ and the untrained classifier $f(x; \theta)$.
Output Fully labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$ and trained classifier $f(x; \theta)$

$n \leftarrow$ Desired dataset length
$q \leftarrow$ Query-pool size
$Q(x) \leftarrow$ Query Function
$T \leftarrow$ Number of SFP's
while $\mathcal{L}$ length $< n$ **do**
    Retrain $f(x; \theta)$ on $\mathcal{L}$
    $P \leftarrow \emptyset$
    for $t = 0, ..., T$ **do**
        insert $f(\mathcal{U}; \theta_t)$ into P
    end for
    Sort $\mathcal{U}$ based on $Q(P)$
    Let Oracle assign labels to $\mathcal{U}_0^q$
    Insert $\mathcal{U}_0^q$ into $\mathcal{L}$
    Remove $\mathcal{U}_0^q$ from $\mathcal{U}$
end while

Algorithm 3 The AL loop with Redundancy Elimination by Training (RET).

Input Labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$, the unlabeled data $\mathcal{U} = \{(x_i, \emptyset)\}_i^n$ and the untrained classifier $f(x; \theta)$.

Output Fully labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$ and trained classifier $f(x; \theta)$

$n \leftarrow$ Desired dataset length
$r \leftarrow$ Redundancy-pool size
$q \leftarrow$ Query-pool size
$T \leftarrow$ Number of SFP's
$Q(x) \leftarrow$ Query Function
while $\mathcal{L}$ length $< n$ **do**
  Retrain $f(x; \theta)$ on $\mathcal{L}$
  $P \leftarrow \emptyset$
  for $t = 0, ..., T$ **do**
    insert $f(\mathcal{U}; \theta_t)$ into P
  end for
  Sort $\mathcal{U}$ based on $Q(P)$
  $U \leftarrow \emptyset$
  $queried \leftarrow 0$
  while $queried < q$ **do**
    for $t = 0, ..., T$ **do**
      insert $f(\mathcal{RP}; \theta_t)$ into U
    end for
    $i \leftarrow argmin(U)$
    Let Oracle assign label to $\mathcal{U}_i$
    Train $f(x; \theta)$ on $\mathcal{U}_i$
    Insert $\mathcal{U}_i$ into $\mathcal{L}$
    Remove $\mathcal{U}_i$ from $\mathcal{U}$
    $queried \leftarrow queried + 1$
  end while
end while

Algorithm 4 The AL loop with Redundancy Elimination by Cosine Similarity (RECS).

Input Labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$, the unlabeled data $\mathcal{U} = \{(x_i, \emptyset)\}_i^n$ and the untrained classifier $f(x; \theta)$.
Output Fully labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$ and trained classifier $f(x; \theta)$

$n \leftarrow$ Desired dataset length
$u \leftarrow$ Redundancy-pool size
$q \leftarrow$ Query-pool size
$l \leftarrow$ Cosine similarity threshold
$T \leftarrow$ Number of SFP's
$Q(x) \leftarrow$ Query Function
$Cos(x, y) \leftarrow$ Cosine similarity between x and y
while $\mathcal{L}$ length $< n$ **do**
  Retrain $f(x; \theta)$ on $\mathcal{L}$
  $P \leftarrow \emptyset$
  for $t = 0, ..., T$ **do**
    insert $f(\mathcal{U}; \theta_t)$ into P
  end for
  Sort $\mathcal{U}$ based on $Q(P)$
  $U \leftarrow \emptyset$
  while $Ulength < q$ **do**
    for $i = 0, ..., u$ **do**
      if $Cos(\mathcal{U}_i, U_0^{Ulength}) < l$ **then**
        insert $\mathcal{U}_i$ into U
      end if
    end for
    $l \leftarrow l - 0.01$
  end while
  Reset l to initial value
  Let Oracle assign labels to U
  Insert U into $\mathcal{L}$
  Remove U from $\mathcal{U}$
end while

Algorithm 5 The AL loop with SUD.

Input Labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$, the unlabeled data $\mathcal{U} = \{(x_i, \emptyset)\}_i^n$ and the untrained classifier $f(x; \theta)$.
Output Fully labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$ and trained classifier $f(x; \theta)$

$n \leftarrow$ Desired dataset length
$q \leftarrow$ Query-pool size
$k \leftarrow$ Amount of similar examples to compute density with
$T \leftarrow$ Number of SFP's
$Q(x) \leftarrow$ Query Function
$Cos(x, y) \leftarrow$ Cosine similarity between x and y
while $\mathcal{L}$ length $< n$ **do**
  Retrain $f(x; \theta)$ on $\mathcal{L}$
  $P \leftarrow \emptyset$
  $E \leftarrow \emptyset$
  for $t = 0, ..., T$ **do**
    Insert $f(\mathcal{U}; \theta_t)$ into P
  end for
  for *example* in $\mathcal{U}$ **do**
    $similar \leftarrow Sort(Cos(example, U))$
    Insert $\frac{sum(similar_0^k))}{k}$ into E
  end for
  Sort $\mathcal{U}$ based on $Q(P{*}E)$
  Let Oracle assign labels to $\mathcal{U}_0^q$
  Insert $\mathcal{U}_0^q$ into $\mathcal{L}$
  Remove $\mathcal{U}_0^q$ from $\mathcal{U}$
end while

Matrix Completion using Regularised Matrix Factorisation*

Abdolrahman Khoshrou[1[0000−0002−8545−4591]] and
Eric J. Pauwels[1[0000−0001−7518−6856]]

Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands
{khoshru,eric.pauwels}@cwi.nl
www.cwi.nl/research/groups/intelligent-and-autonomous-systems

Abstract. Singular Value Decomposition (SVD) and Principal Component Analysis (PCA), are well-known linear matrix decomposition techniques that are widely used in machine learning with applications such as dimension reduction and clustering. However, SVD/PCA is sensitive to noise in the input data. We show how different formulations of the regularisation functional lead to qualitatively different solutions.

Keywords: Singular Value Decomposition (SVD) · Principal Component Analysis (PCA) · matrix completion and factorisation · regularisation · graph Laplacian · manifold learning.

1 Introduction and Related Work

Singular Value Decomposition (SVD) and its close relative, Principal Component Analysis (PCA), are well-known linear matrix factorisation techniques that are widely used in machine learning and artificial intelligence with applications as varied as dimension reduction and clustering, matrix completion [1] (e.g. for recommender systems), dictionary learning [2] and time series analysis [3].

In their abstract version, SVD and PCA amount to two different but related types of matrix factorisation. More precisely, given a general (data) matrix A, the aim is to approximate it as a product of lower-dimensional matrices. Specifically:

- PCA-type decomposition: $A \approx PQ^T$ where the columns of Q are orthonormal, i.e. $Q^TQ = I$;
- SVD-type decomposition: $A \approx PBQ^T$ where B is diagonal, while $P^TP = I$ and $Q^TQ = I$.

The approximation in the above equations is measured in terms of the Frobenius (matrix) norm which for an arbitrary matrix $X \in \mathbb{R}^{n\times m}$ is defined as:

$$\|X\|^2 = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}^2 = \mathrm{Tr}(XX^T) = \mathrm{Tr}(X^TX) = \|X^T\|^2. \tag{1}$$

* The authors acknowledge partial support by Dutch NWO ESI-Bida project NEAT (647.003.002).

Although these factorisations are both conceptually simple and effective, it is well-known that they are sensitive to noise and outliers in the input data. As a consequence, some modifications of the original algorithms have been proposed to alleviate the effect of such disturbances [5, 6]. Candes et al. [7] introduce *Robust PCA (RPCA)* which aims to separate signal from outliers by decomposing any given matrix into the sum of a low-rank approximation and a sparse matrix of outliers. An extension of this work for inexact recovery of the data is presented in [10]. Another example of sparse PCA using low rank approximation is proposed in [11]. Adding a regularisation term is another versatile way to tackle the problem of noisy input. For instance, Dumitrescu et al. [12] show how a regularized version of K-SVD algorithm can be adapted to the Dictionary Learning (DL) problem. However, the presence of noise in the input is not the only reason to invoke regularisation. Recent research [13] shows that in many real data sets, not only the observed data, but features also lie on a (non-)linear low dimensional manifold. He et al. [8] consider a setup where the columns of the matrix A are interpreted as data points, then the rows are features. The neighbourhood structure of both the data points and the features give rise to distinct graphs (so-called data and feature graphs) and to their corresponding graph Laplacians (L_d and L_f respectively). The resulting regularised PCA is referred to as the *graph-dual Laplacian PCA* (gDLPCA) and for a given data matrix A, is obtained by minimising the below functional:

$$J(V,Y) = \|A - VY\|^2 + \alpha \operatorname{Tr}(V^T L_d V) + \beta \operatorname{Tr}(Y L_f Y^T) \qquad \text{s.t. } V^T V = I \quad (2)$$

The ability of the graph dual regularization technique to incorporate both data and feature structure has deservedly attracted considerable attention in dimensionality reduction applications [4, 8, 14]. In the present paper, we take the functional (2) as a starting point and investigate the two factorisation approaches mentioned above (invoking eq. (1) to recast the trace as a norm):

- PCA-type decomposition ($A \approx PQ^T$) by minimising the regularisation functional:
$$\|A - PQ^T\|^2 + \lambda \|DP\|^2 + \mu \|GQ\|^2 \quad (3)$$
- SVD-type decomposition ($A \approx PBQ^T$) by minimising the regularisation functional:
$$\|A - PBQ^T\|^2 + \lambda \|DP\|^2 + \mu \|GQ\|^2 \quad (4)$$

The minimisation of the functional (3) is discussed in [8], but the proposed solution contains an error which we correct in this paper. In addition, we also provide an algorithm to solve functional (4), which somewhat surprisingly is quite different from the one for (3).

The rest of this paper is organised as follows: We finalise this section by recapitulating some important facts about SVD. In Sect. 2 and 3 we derive an algorithm for minimisation of the regularised version of PCA- and SVD-type factorisation, respectively. In Sect. 4 we show how gradient descent can be implemented by drawing on some elementary facts from Lie-group theory. Finally, we conclude this paper by giving some pointers to potential extensions.

For the sake of completeness, we first recall the well-known SVD result; for more details we refer to standard textbooks such as [15].

Theorem 1 (Singular Value Decomposition, SVD). *Any real-valued $n \times m$ matrix A can be factorized into the product of three matrices:*

$$A = USV^T \tag{5}$$

where $U \in \mathcal{O}(n)$ and $V \in \mathcal{O}(m)$ are orthonormal and S is an $n \times m$ diagonal matrix where the elements on the main "diagonal" (so-called singular values) are non-negative (i.e. $\sigma_i := S_{ii} \geq 0$ for $1 \leq i \leq \min(n, m)$). Assuming that the rank $rk(A) = r \leq \min(n, m)$, we can sort the singular values such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0 = \sigma_{r+1} = \ldots = \sigma_{\min(n,m)}$ and recast eq. (5) as

$$A = \sum_{i=1}^{r} \sigma_i U_i V_i^T \tag{6}$$

where U_i, V_i are the i-th columns of U and V, respectively. For the singular vectors, we introduce the short-hand notation $U_{(1:k)} := [U_1, U_2, \ldots, U_k]$ and $V_{(1:k)} := [V_1, V_2, \ldots, V_k]$ to denote the matrix comprising the first k columns of U and V, respectively. In this notation, eq. (6) can be expressed concisely as:

$$A = U_{(1:r)}\, diag(\sigma_1, \ldots, \sigma_r)\, V_{(1:r)}^T \tag{7}$$

□

To appreciate the significance of Theorem 1, it is helpful to highlight its geometric interpretation. Recall that any $n \times m$ matrix A gives rise to a corresponding linear transformation $A : \mathbb{R}^m \to \mathbb{R}^n$ that maps the standard basis in $\mathbb{R}^m$ into the columns of A, i.e., $A\mathbf{e}_k = A_k$ where $\mathbf{e}_k = (0, 0, \ldots, 0, 1, 0, \ldots, 0)^T$ is a column vector. Roughly speaking, the SVD theorem therefore tells us that it is always possible to select an *orthonormal* basis in $\mathbb{R}^m$ (columns of V) that is mapped (up to non-negative scaling factors, i.e. the singular values) into an *orthonormal* basis in $\mathbb{R}^n$ (columns of U). This is immediately obvious from eq. (6):

$$AV_\ell = \sum_{k=1}^{r} \sigma_k U_k V_k^T V_\ell = \sum_{k=1}^{r} \sigma_k U_k \delta_{k\ell} = \sigma_\ell U_\ell$$

where $\delta_{k\ell}$ is a Kronecker delta function. It is worth noting that insisting on the orthogonality of V ($V^T V = I$) is not restrictive. Indeed, a linear transformation is completely and uniquely determined by specifying its effect on any basis, and there is no loss of generality by insisting on the orthonormality of this basis. However, the non-trivial message of this theorem is that this orthonormal basis (V) can be chosen in such a way that its image U under A is *also orthonormal* (again, up to non-negative scalings). Furthermore, in a generic case (where all singular values are different) the singular value decomposition is unique, up to an arbitrary relabeling of the *basis-vectors* and a simultaneous sign-flip of corresponding columns in U and V, i.e. $(U_\ell, V_\ell) \to (-U_\ell, -V_\ell)$ for any number

of columns. The importance of the SVD result, and the starting point for this paper, is the following well-known minimisation result (more details can be found in [16]).

Theorem 2 (Eckart-Young-Mirsky Theorem: Optimal low rank approximation). *Let us consider an $n \times m$ matrix A with rank $rk(A) = r \leq \min(n, m)$. For $k < r$, finding the rank-k matrix A_k that is closest to A in (Frobenius) norm gives rise to the following constrained minimisation problem:*

$$\min_{A_k} \|A - A_k\|^2 \qquad s.t. \quad rk(A_k) \leq k.$$

The solution to this problem is obtained by truncating the SVD expansion eq. (6) *after the k-th largest singular value:*

$$A_k = \sum_{i=1}^{k} \sigma_i U_i V_i^T = U_{(1:k)}\, diag(\sigma_1, \ldots, \sigma_k)\, V_{(1:k)}^T. \tag{8}$$

□

Recall that a rank-k matrix of size $n \times m$ can always be written as a product $A_k = PQ^T$ where $P \in \mathbb{R}^{n\times k}$ and $Q \in \mathbb{R}^{m\times k}$ are matrices of full rank k. Again, in this factorisation, there is no loss of generality in requiring $Q^TQ = I_k$. In fact, it is necessary to remove indeterminacy due to arbitrary but trivial rescalings such as $P \longmapsto rP$ while $Q \longmapsto (1/r)Q$ (with $r \neq 0$), and the like. Hence, one can reformulate Theorem 2 as the factorisation result in Theorem 3.

Theorem 3 (PCA-type factorisation). *Assume that the $n \times m$ matrix A has rank $rk(A) = r \leq \min(n, m)$. We now define the functional $G(P, Q)$ as follow:*

$$G(P, Q) = \|A - PQ^T\|^2 \tag{9}$$

and the corresponding constrained optimisation problem:

$$\min_{P,Q} G(P, Q) \quad s.t. \qquad Q^TQ = I_k, \quad and \quad k < r$$

A solution to the above constrained minimisation problem (in $P \in \mathbb{R}^{n\times k}$ and $Q \in \mathbb{R}^{m\times k}$) is given by (using the SVD notation given above):

$$Q = V_{(1:k)} \qquad and \qquad P = U_{(1:k)}\, diag(\sigma_1, \ldots, \sigma_k) \tag{10}$$

hence:

$$PQ^T = \sum_{i=1}^{k} \sigma_i U_i V_i^T. \tag{11}$$

□

From (10) it also follows that P^TP is diagonal, but not necessarily equal to the identity. Note that if we drop the insistence on the diagonal form for P^TP (i.e. P need no longer be an orthogonal frame), then the solution is no longer unique. Indeed, for any $k \times k$ orthogonal matrix R with $R^TR = I_k = RR^T$, it is clear that $P' = PR$ and $Q' = QR$ are also solutions. In this case: $Q'^TQ' = R^TQ^TQR = I_k$ but $P'^TP' = R^TP^TPR = R^T(SS^T)R$ is in general a positive definite symmetric matrix.

2 Regularisation for PCA-type factorisation

The following theorem outlines an obvious generalisation to regularised version of the minimisation problem.

Theorem 4 (Regularised PCA). *Let A be an $n \times m$ matrix of rank $r \leq \min(n, m)$. For $k \leq r$, let $P \in \mathbb{R}^{n\times k}$ and $Q \in \mathbb{R}^{m\times k}$ full rank matrices (i.e. of rank k). Furthermore, for arbitrary (non-zero) integers d and g we introduce regularisation matrices $D \in \mathbb{R}^{d\times n}$ and $G \in \mathbb{R}^{g\times m}$, as well as weights $\lambda, \mu \geq 0$. We now define the following functional F in the variables P and Q:*

$$F(P, Q) = \|A - PQ^T\|^2 + \lambda \|DP\|^2 + \mu \|GQ\|^2 \tag{12}$$

and pose the corresponding constrained optimisation problem:

$$\min_{P,Q} F(P, Q) \qquad s.t. \qquad Q^T Q = I_k. \tag{13}$$

Introducing short-hand notation $L := D^T D \in \mathbb{R}^{n\times n}$ and $M := G^T G \in \mathbb{R}^{m\times m}$ (both symmetric and positive semi-definite), the solution to the constrained optimisation problem (13) *is constructed as follows:*

- *The k columns of the $m \times k$ matrix Q are the eigenvectors of the $m \times m$ matrix:*
$$K := A^T (I_n + \lambda L)^{-1} A - \mu M$$
corresponding to the k largest eigenvalues;
- *Furthermore: $P = (I_n + \lambda L)^{-1} AQ$*

For the sake of completeness, we reiterate that the condition $Q^T Q = I_k$ is not restrictive but necessary to eliminate arbitrary rescalings. In passing, we point out that result above corrects an error in [8] where it is incorrectly stated that $P = AQ$.

Proof. Since the variable P in the functional (12) in unconstrained, we can identify the optimum in P (for fixed Q) by computing the gradient:

$$\frac{1}{2}\nabla_P F = (PQ^T - A)Q + \lambda D^T DP \tag{14}$$

and solving for P:

$$\nabla_P F = 0 \quad \Rightarrow \quad P\underbrace{Q^T Q}_{I_k} - AQ + \lambda LP = 0 \quad \Rightarrow \quad (I_k + \lambda L)P = AQ. \tag{15}$$

This condition needs to hold at the solution point. By first re-writing $F(P, Q)$ formula as the trace of matrices and then plugging in (15), we have:

$$\begin{aligned} F(P, Q) &= \operatorname{Tr}\left[(A - PQ^T)(A^T - QP^T)\right] + \lambda \operatorname{Tr}(P^T LP) + \mu \operatorname{Tr}(Q^T MQ) \\ &= \operatorname{Tr}\left[AA^T - AQP^T - PQ^T A^T + PQ^T QP^T\right] + \lambda \operatorname{Tr}(P^T LP) + \mu \operatorname{Tr}(Q^T MQ) \end{aligned}$$

Considering the fact that the trace operator is invariant under transposition as well as cyclic permutation, and plugging in eq. (15) we arrive at:

$$\begin{aligned} F(P,Q) &= \mathrm{Tr}\left[AA^T - 2(I_n+\lambda L)PP^T + PP^T\right] + \lambda\,\mathrm{Tr}(P^TLP) + \mu\,\mathrm{Tr}(Q^TMQ) \\ &= \mathrm{Tr}\left(AA^T - PP^T - 2\lambda LPP^T\right) + \lambda\,\mathrm{Tr}(P^TLP) + \mu\,\mathrm{Tr}(Q^TMQ) \\ &= \mathrm{Tr}(AA^T) - \mathrm{Tr}(PP^T) - 2\lambda\,\mathrm{Tr}(LPP^T) + \lambda\,\mathrm{Tr}(P^TLP) + \mu\,\mathrm{Tr}(Q^TMQ) \\ &= \mathrm{Tr}(AA^T) - \mathrm{Tr}(P^TP) - \lambda\,\mathrm{Tr}(P^TLP) + \mu\,\mathrm{Tr}(Q^TMQ) \\ &= \mathrm{Tr}(AA^T) - \mathrm{Tr}\left[P^T(I_n+\lambda L)P\right] + \mu\,\mathrm{Tr}(Q^TMQ). \end{aligned} \tag{16}$$

Extracting P and its transpose from eq. (15):

$$P = (I_n+\lambda L)^{-1}AQ \quad \Rightarrow \quad P^T = Q^TA^T(I_n+\lambda L)^{-1} \tag{17}$$

we arrive at:

$$F(P,Q) = \mathrm{Tr}(AA^T) - \mathrm{Tr}\left[Q^T\left(A^T(I_n+\lambda L)^{-1}A - \mu M\right)Q\right]. \tag{18}$$

Therefore, in order to minimize F, one must maximize the right-most term as $\mathrm{Tr}(AA^T)$ is a constant. This is achieved by selecting for Q, eigenvectors corresponding to the k largest eigenvalues of $(A^T(I_n+\lambda L)^{-1}A - \mu M)$. Once Q is determined, P is obtained via eq. (17). □

As a concluding remark, we point out that the matrix $I_n + \lambda L$ is always invertible. Indeed, since $L = D^TD$ is positive semi-definite and symmetric, it has a complete set of eigenvectors with corresponding non-negative eigenvalues, i.e., $L = W\Lambda W^T$, where W is orthogonal (i.e. $W^TW = WW^T = I_n$) and $\Lambda \geq 0$. Hence, the matrix $(I_n+\lambda L)$ has strictly positive diagonal elements, and is indeed invertible. Some illustrative numerical experiments can be found in [17].
Some special cases:

- $\boxed{\lambda = 0 \text{ and } \mu = 0}$: In that case, Q comprises the first k eigenvectors of $K = A^TA$ and $P = AQ$, which means that we end up with standard SVD, as expected. Some numerical experiments can be found in [17].
- $\boxed{D = I_n \text{ and } \mu = 0}$: The following section provides an overview of the results in [12] where a regularized K-SVD problem is addressed. In the aforementioned paper, the authors consider a special case, where $\mu = 0$ and $D = I_n$. Since this implies that $L = D^TD = I_n$ and $\mu M = 0$, the matrix K simplifies to $K = \frac{1}{1+\lambda}A^TA$. The eigenvectors of K are therefore the right singular vectors of A (i.e. the eigenvectors of A^TA). Hence $Q = V_{(1:k)}$, and as a result: $P = \frac{1}{1+\lambda}AQ \quad and \quad AQ = U_{(1:k)}\,diag(\sigma_1,\ldots,\sigma_k)$. In particular, for $k = 1$ (the rank-1 reconstruction), we obtain: $Q = \mathbf{v}_1$ and $P = \frac{\sigma_1}{1+\lambda}\mathbf{u}_1$ which is the result that can be found in [12]. The experiments are available in [17].

3 Regularisation for SVD-type factorisation

We now turn our attention to the SVD-type factorisation which looks for an approximation of the form:

$$A \approx PBQ^T \quad \text{s.t.} \quad Q^TQ = I_k, \quad \forall i \in \{1, 2, \ldots, k\} : \|P_i\| = 1, \quad \text{and } B \text{ diagonal.}$$

Since the columns of P and Q are of unit length, they only pin down the *structure* of A, whereas the diagonal matrix $B = diag(\beta_1, \beta_2, \ldots, \beta_k)$ captures the *amplitude* of the corresponding structures. Similar to before, the columns of Q are orthonormal (i.e., $Q^TQ = I_k$). However, unlike before, the columns of P are now only required to have unit length. In light of the aforementioned SVD-type matrix factorisation technique, Theorems 5 and 6 provide an alternative solution to the lower dimensional matrix approximation problem. Theorem 5 first addresses the simplified case for $\mu = 0$, but we return to the general case in Theorem 6.

Theorem 5 (Regularised SVD). *Let A be an $n \times m$ matrix of rank $r \leq \min(n, m)$. For $k \leq r$, let $P \in \mathbb{R}^{n\times k}$ and $Q \in \mathbb{R}^{m\times k}$ of rank k, while $B \in \mathbb{R}^{k\times k}$ is diagonal (i.e. $B = diag(\beta_1, \beta_2, \ldots, \beta_k)$) where all the off-diagonal elements of B are zero. Furthermore, for arbitrary non-zero integer d we introduce regularisation matrix $D \in \mathbb{R}^{d\times n}$, as well as weight $\lambda \geq 0$. Finally, we introduce the short-hand notation $L := D^TD \in \mathbb{R}^{n\times n}$ (symmetric and positive-definite). We are now in a position to define the following functional F in the variables P, Q and B:*

$$F(P, Q, B) = \|A - PBQ^T\|^2 + \lambda \|DP\|^2 \tag{19}$$

and the corresponding constrained optimisation problem:

$$\min_{P,Q,B} F(P, Q, B) \qquad \textit{s.t.}$$
$$Q^TQ = I_k, \ \|P_i\| = 1 \ \forall i \in \{1, 2, \ldots, k\}, \quad \textit{and } B \textit{ diagonal.} \tag{20}$$

Algorithm 1 in below proposes a solution to this problem.

Proof. Since B is unconstrained, we can determine its optimal value by computing the derivative with respect to B and equating it to zero:

$$\nabla_B F(P, Q, B) = \nabla_B \|A - PBQ^T\|^2. \tag{21}$$

Expanding the norm in terms of a trace (cf. eq. (1)), and using the invariance of a trace under transposition, we arrive at (recall $Q^TQ = I_k$):

$$\begin{aligned}
\|A - PBQ^T\|^2 &= \operatorname{Tr}\left[(A - PBQ^T)(A^T - QBP^T)\right] \\
&= \|A\|^2 - 2\operatorname{Tr}(P^TAQB) + \operatorname{Tr}(B^2P^TP) \\
&= \|A\|^2 - 2\sum_{i=1}^{k}(P^TAQ)_{ii}\,\beta_i + \sum_{i=1}^{k}(P^TP)_{ii}\,\beta_i^2 \\
&= \|A\|^2 - 2\sum_{i=1}^{k}(P^TAQ)_{ii}\,\beta_i + \sum_{i=1}^{k}\beta_i^2
\end{aligned} \tag{22}$$

Hence, the gradient of the functional F with respect to B is obtained as follow:

$$\frac{\partial}{\partial \beta_i} \|A - PBQ^T\|^2 = 2\,(\beta_i - (P^T AQ)_{ii}).$$

For given P and Q, we find the optimal B when the gradient vanishes:

$$\beta_i = (P^T AQ)_{ii} \qquad \forall i \in \{1, 2, \ldots, k\}. \tag{23}$$

Plugging this optimal choice back into eq. (22) the functional (19) simplifies to

$$\|A - PBQ^T\|^2 = \|A\|^2 - \sum_{i=1}^{k} \beta_i^2 \tag{24}$$

To recast eq. (24) in terms of P and Q (in order to eliminate B), we observe that for every element of an arbitrary matrix H we have $H_{ij} = \mathbf{e}_i^T H \mathbf{e}_j$, where $\mathbf{e}_i = (0, 0, \ldots, 1, \ldots, 0)^T$ are the standard basis vectors. Hence, using the fact that the diagonal of a matrix is unchanged under transposition, we conclude that

$$\beta_i = \begin{cases} (P^T AQ)_{ii} & = \mathbf{e}_i^T P^T AQ\, \mathbf{e}_i = \mathbf{p}_i^T A \mathbf{q}_i \\ (Q^T A^T P)_{ii} & = \mathbf{e}_i^T Q^T A^T P\, \mathbf{e}_i = \mathbf{q}_i^T A^T \mathbf{p}_i \end{cases}$$

where $\mathbf{p}_i, \mathbf{q}_i$ are the i-th columns of P and Q, respectively. Hence:

$$\sum_{i=1}^{k} \beta_i^2 = \sum_{i=1}^{k} \mathbf{p}_i^T A \mathbf{q}_i\, \mathbf{q}_i^T A^T \mathbf{p}_i. \tag{25}$$

As a final step, we introduce $L = D^T D$ to recast the regularisation term as:

$$\|DP\|^2 = \mathrm{Tr}(P^T LP) = \sum_{i=1}^{k} \mathbf{e}_i^T P^T LP\, \mathbf{e}_i = \sum_{i=1}^{k} \mathbf{p}_i^T L\, \mathbf{p}_i. \tag{26}$$

Plugging eqs. (25) and (26) into eq. (19), we obtain the following simplified form for the functional F (assuming that we eliminate B by using its optimal value):

$$\begin{aligned} F(P, Q) &= \|A\|^2 + F_1(P, Q), \quad \text{where} \\ F_1(P, Q) &= \sum_{i=1}^{k} \mathbf{p}_i^T (\lambda L - A\mathbf{q}_i \mathbf{q}_i^T A^T) \mathbf{p}_i. \end{aligned} \tag{27}$$

Introducing the notation $S(\mathbf{q}) := \lambda L - A\mathbf{q}\mathbf{q}^T A^T$, we conclude that

$$F_1(P, Q) = \sum_{i=1}^{k} \mathbf{p}_i^T S(\mathbf{q}_i) \mathbf{p}_i.$$

Since each $S(\mathbf{q})$ is a symmetric matrix, it can be diagonalised with respect to an orthonormal basis, i.e. there is an orthogonal $n \times n$ matrix W (with

$W^TW = WW^T = I_n$) and a diagonal matrix $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$ (ordered $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$), both depending on $\mathbf{q}$ such that $S(\mathbf{q}) = W(\mathbf{q})\Lambda(\mathbf{q})W(\mathbf{q})^T$ i.e. the columns of W are the eigenvectors of $S(\mathbf{q})$, with the corresponding eigenvalues on the diagonal of Λ. By introducing the notation $\lambda_1(S(\mathbf{q}))$ to denote the smallest eigenvalue of $\Lambda(\mathbf{q})$, we obtain the minimal value $\mathbf{p}_i^T S(\mathbf{q}_i)\mathbf{p}_i = \lambda_1(\mathbf{q}_i)$ when choosing $\mathbf{p}_i$ to be the (unit) eigenvector ($W_1(\mathbf{q}_i)$) corresponding to the smallest eigenvalue. As a consequence, the solution strategy boils down to steps in Algorithm 1. This choice of P, Q and B solves the constrained minimisation problem (20). □

Notice that since P and B are determined after finding Q, this optimisation problem can essentially be translated into a search in the space of Q matrices. Some illustrative numerical experiments are available at [17]. We conclude this section by giving a slightly more general version ($\mu \neq 0$) of the previous theorem, thus re-establishing the symmetry between P and Q.

Algorithm 1: Proposed RSVD method ($\mu = 0$)

Input: A, k, λ, D
Output: P, B, Q
Initialization
while *no convergence* **do**

1. Determine the $m \times k$ matrix $Q = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k]$ (with orthonormal columns: $Q^TQ = I_k$) such that the sum of the smallest eigenvalue of each of the k symmetric matrices $S(\mathbf{q}_i)$ is minimal, i.e.:

$$\min_Q \psi(Q) = \min_Q \sum_{i=1}^{k} \lambda_1(\mathbf{q}_i) \qquad \text{such that } Q^TQ = I_k$$

   where $\lambda_1(\mathbf{q}_i) = \min(eig(S(\mathbf{q}_i))$. To this end we use gradient descent (see Sect. 4).
2. For each $\mathbf{q}_i$ as determined above, take $\mathbf{p}_i$ to be the eigenvector $W_1(\mathbf{q}_i)$ corresponding to the smallest eigenvector $\lambda_1(\mathbf{q}_i)$. Construct the $n \times k$ matrix $P = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k]$.
3. Finally, set $B = diag(\beta_1, \ldots, \beta_n)$ where $\beta_i = (P^TAQ)_{ii}$.

end

Theorem 6 (Regularised SVD, symmetric version). *Let A be an $n \times m$ matrix of rank $r \leq \min(n, m)$. For $k \leq r$, let $P \in \mathbb{R}^{n\times k}$ and $Q \in \mathbb{R}^{m\times k}$ of rank k, while $B \in \mathbb{R}^{k\times k}$ diagonal (i.e. $B = diag(\beta_1, \beta_2, \ldots, \beta_k)$). Furthermore, for arbitrary non-zero integers d and g we introduce regularisation matrices $D \in \mathbb{R}^{d\times n}$, and $G \in \mathbb{R}^{g\times m}$, as well as weights $\lambda, \mu \geq 0$. Finally, we introduce the short-hand notation $L := D^TD \in \mathbb{R}^{n\times n}$ and $M := G^TG \in \mathbb{R}^{m\times m}$ symmetric*

and positive-definite). We are now in a position to define the following functional F in the variables P, Q and B:

$$F(P,Q,B) = \|A - PBQ^T\|^2 + \lambda \|DP\|^2 + \mu \|GQ\|^2 \tag{28}$$

and the corresponding constrained optimisation problem:

$$\min_{P,Q,B} F(P,Q,B) \quad s.t.$$
$$Q^TQ = I_k, \quad \|P_i\| = 1,\ \forall i \in \{1,2,\ldots,k\} \quad \textit{and } B \textit{ diagonal.} \tag{29}$$

Algorithm 2 provides a solution to the aforementioned problem.

Proof. Using the notation introduced above and in Theorem 5, we see that

$$\|GQ\|^2 = \mathrm{Tr}(Q^TMQ) = \sum_{i=1}^{k} \mathbf{q}_i^T M \mathbf{q}_i.$$

Hence, the functional (28) can be recast as:

$$F(P,Q) = \|A\|^2 + F_2(P,Q), \qquad \text{where}$$
$$F_2(P,Q) = \sum_{i=1}^{k} \mathbf{p}_i^T(\lambda L - A\mathbf{q}_i\mathbf{q}_i^TA^T)\mathbf{p}_i + \mu \sum_{i=1}^{k} \mathbf{q}_i^T M \mathbf{q}_i. \tag{30}$$

The minimum of each term in the first summation in F_2 is equal to the smallest eigenvalue $\lambda_1(S(\mathbf{q}_i))$. Finding the minimum for the constrained optimisation problem (29) therefore amounts to finding the minimum of the functional:

$$\psi(Q) := \sum_{i=1}^{k} \left(\lambda_1(S(\mathbf{q}_i)) + \mu\, \mathbf{q}_i^T M \mathbf{q}_i\right) \tag{31}$$

subject to the constraint $Q^TQ = I_k$. Therefore, the minimisation problem again calls for a minimisation in Q space, as the optimal choice for P (corresponding eigen-vectors) follows automatically. We therefore arrive at the solution detailed in Algorithm 2. Some illustrative numerical examples are available in [17]. □

4 Computational Aspects

From Algorithm 2, it becomes clear that full regularisation problem can be reduced to a simpler constrained minimisation problem (31). Since the ψ-functional is smooth on a compact domain, the minimum is guaranteed to exist and one can use gradient descent to locate it. However, gradient descent needs to respect the constraint $Q^TQ = I_k$. This is achieved by applying orthogonal transformations to the current Q matrix, as it will preserve orthonormality. Specifically, recall

Algorithm 2: Proposed RSVD method ($\mu \neq 0$)

Input: A, k, μ, λ, D, G
Output: P, B, Q
Initialization
while *no convergence* **do**

1. For any unit vector $\mathbf{q} \in \mathbb{R}^m$ we define $S(\mathbf{q}) = \lambda L - Aqq^T A^T$.
   Since this is a symmetric $n \times n$ matrix, it has a complete set of eigenvectors and corresponding eigenvalues.
   Denote the smallest eigenvalue of each $S(\mathbf{q}_i)$ as $\lambda_1(S(\mathbf{q}_i))$.
2. For a given $m \times k$ matrix $Q = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k]$
   (with orthonormal columns: $Q^T Q = I_k$) compute the functional:

$$\psi(Q) := \sum_{i=1}^{k} \Big(\lambda_1(S(\mathbf{q}_i)) + \mu\, \mathbf{q}_i^T M \mathbf{q}_i \Big)$$

   and use gradient descent (on the compact *torus domain*, see Sect. 4) to find the minimum.
3. For each $\mathbf{q}_i$ as determined above, take $\mathbf{p}_i$ to be the eigenvector $W_1(\mathbf{q}_i)$ corresponding to the smallest eigenvector $\lambda_1(S(\mathbf{q}_i))$.
   Construct the $n \times k$ matrix $P = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k]$.
4. Finally, set $B = diag(\beta_1, \ldots, \beta_n)$ where $\beta_i = (P^T AQ)_{ii}$.

end

that any orthogonal $m \times m$ matrix R with determinant 1 (rather than -1) can be generated by exponentiating an appropriate skew-symmetric matrix K [19]:

$$R = \exp(tK) \equiv I_m + tK + \frac{1}{2!}t^2K^2 + \ldots + \frac{1}{n!}t^nK^n + \ldots \quad (\text{with } K^T = -K)$$

By choosing t sufficiently small, the orthogonal transformation is close to the identity I_m. Furthermore, we can restrict the variations to orthogonal transformations that result from exponentiating a basis for the space of skew-symmetric matrices. Such a basis is provided by the $m(m-1)/2$ skew-symmetric matrices $K^{(ij)}$ (where $1 \leq i < j \leq m$) that has two non-zero entries: $K_{ij}^{(ij)} = 1, K_{ji}^{(ij)} = -1$. Given the current value Q_0, we construct nearby values for Q by looping over $K^{(12)}, K^{(13)}, K^{(23)}, \ldots etc$ and constructing the corresponding orthogonal matrices $R_{ij}(t) = \exp(tK^{(ij)})$. Denoting these "*infinitesimal*" rotation matrices as R_α (where $\alpha = 1, \ldots, m(m-1)/2$), we see that the partial derivatives with respect to these rotations can be estimated as:

$$\frac{\partial \psi(Q)}{\partial R_\alpha} \approx \frac{\psi(R_\alpha(t)Q_0) - \psi(Q_0)}{t} \qquad (\text{for } t \text{ sufficiently small}).$$

From these results we can select the infinitesimal rotation that results in the steepest descent. Since computing ψ is computationally expensive (it requires determining eigenvalues), a viable alternative to computing the gradient is ran-

dom descent: generate random rotations (by exponentiating random skew symmetric matrices) and check whether they result in a lower ψ-value. As soon as one is found, proceed in that direction, and repeat the process.

We conclude this section with a concrete example: smoothing a noisy matrix. We start from the assumption (cf., [8,13,18]) that the $n \times m$ data matrix A has a relatively smooth underlying structure that is corrupted by noise: $A = \mathbf{u}\mathbf{v}^T + \tau Z$, where the $n \times m$ matrix Z has independent standard normal entries, and τ controls the size of the noise. To recover the underlying "signals" $\mathbf{u}$ and $\mathbf{v}$, we minimise the SVD-type regularisation functional (28) where the smoothness of the result is enforced by using regularisation matrices D and F that extract the second derivative, i.e. they have the template $(1 \quad -2 \quad 1)$ along the diagonal. A typical result for a rank-1 ($k = 1$) approximation is depicted in Figure 1, and compared to the standard SVD solution. This illustrative example is available in [17].

5 Conclusions and Future Research

Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) are important matrix factorisation techniques that underpin numerous applications. However, disturbances in the input (noise, outliers or missing values) have a significant effect on the outcome. We investigated regularisation in two different but related versions of the factorisation, and detailed the solution algorithms. An important topic for further research would be to find ways in which the gradient descent procedure in Algorithms 1 and 2 can be accelerated by taking advantage of the fact that the functional is very smooth and locally approximately quadratic. In addition, it would be useful to derive some estimates for appropriate values for the weights λ and μ in terms of the noise characteristics of the underlying signal. Finally, although the P matrix in Algorithm 2 has unit-length columns, they are not necessarily orthogonal ($P^T P = I$) as is the case in standard SVD. In fact, numerical experiments seem to indicate that such a constraint is not compatible with minimisation of the functional. This requires further theoretical elucidation.

References

1. Davenport, Mark A. and Romberg, Justin, An Overview of Low-Rank Matrix Recovery From Incomplete Observations, IEEE J. Selected Topics in Signal Proc., vol 10, pp. 608–22 (2016)
2. Tošić, Ivana and Frossard, Pascal, Dictionary learning, IEEE Signal Processing Magazine, volume 28, pp. 27–38 (2011)
3. Khoshrou, Abdolrahman and Pauwels, Eric J, Data-driven pattern identification and outlier detection in time series, Science and Information Conference proceedings, pp. 471–484 (2018)
4. Yin, Ming and Gao, Junbin and Lin, Zhouchen and Shi, Qinfeng and Guo, Yi, Dual graph regularized latent low-rank representation for subspace clustering, IEEE Transactions on Image Processing, volume 24, pp. 4918–4933 (2015)

5. Brooks, J Paul and Dulá, José H and Boone, Edward L, A pure L1-norm principal component analysis, Computational statistics & data analysis, volume 61, pp. 83–98 (2013)
6. Kwak, Nojun, Principal component analysis by $L\_\{p\}$-norm maximization, IEEE Transactions on Cybernetics, volume 44, pp. 594–609 (2013)
7. Candès, Emmanuel J and Li, Xiaodong and Ma, Yi and Wright, John, Robust principal component analysis?, Journal of the ACM (JACM), =volume 58, pp. 1–37 (2011)
8. He, Jinrong and Bi, Yingzhou and Liu, Bin and Zeng, Zhigao, Graph-dual Laplacian PCA, J. Ambient Intelligence and Humanized Computing, vol 10, pp. 3249–62 (2019)
9. Shahid, Nauman and Kalofolias, Vassilis and Bresson, Xavier and Bronstein, Michael and Vandergheynst, Pierre, Robust principal component analysis on graphs, Proceedings of the IEEE International Conference on Computer Vision, pp. 2812–2820 (2015)
10. Zhou, Zihan and Li, Xiaodong and Wright, John and Candes, Emmanuel and Ma, Yi, Stable principal component pursuit, IEEE Int. Symposium on Information Theory, pp. 1518–22 (2010)
11. Shen, Haipeng and Huang, Jianhua Z, Sparse principal component analysis via regularized low rank matrix approximation, Journal of multivariate analysis, vol 99, pp. 1015–34 (2008)
12. Dumitrescu, Bogdan and Irofti, Paul, Regularized k-svd, IEEE Signal Processing Letters, volume 24, pp. 309–313 (2017)
13. Jin, Taisong and Yu, Jun and You, Jane and Zeng, Kun and Li, Cuihua and Yu, Zhengtao, Low-rank matrix factorization with multiple hypergraph regularizer, Pattern Recognition, volume 48, pp. 1011–1022 (2015)
14. Shahid, Nauman and Perraudin, Nathanael and Kalofolias, Vassilis and Puy, Gilles and Vandergheynst, Pierre, Fast robust PCA on graphs, IEEE Journal of Selected Topics in Signal Processing, volume 10, pp. 740–756 (2016)
15. Strang, Gilbert, Introduction to linear algebra, Wellesley-Cambridge Press Wellesley, MA (1993)
16. Golub, Gene H and Van Loan, Charles F, Matrix computations, JHU press (2013)
17. Abdolrahman Khoshrou, Eric J. Pauwels, code: numerical experiments, url : `https://www.dropbox.com/sh/9k0x1q3hltszxh8/AAD0PChQpg6aNCAxzgK0r1p1a?dl=0` (2021)
18. Gavish, Matan and Donoho, David L, The optimal hard threshold for singular values is $4/\sqrt{3}$, IEEE Transactions on Information Theory, volume 60, pp. 5040–5053 (2014)
19. Iserles, Arieh and Munthe-Kaas, Hans Z and Nørsett, Syvert P and Zanna, Antonella, Lie-group methods, Acta numerica, volume 9, pp. 215–365 (2000)

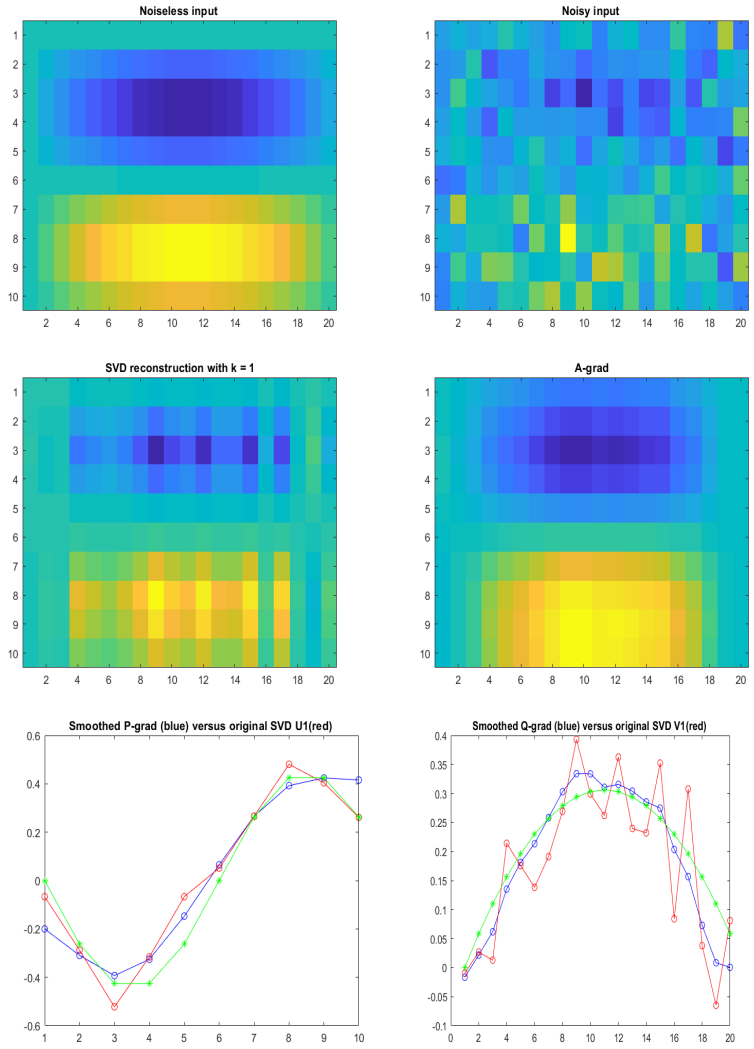


Fig. 1. Reconstruction of noisy matrix based on RSVD. Top left: noise-less rank-1 matrix $\mathbf{u}\mathbf{v}^T$, (image) , top right: noisy input image $\mathbf{u}\mathbf{v}^T + \tau Z$ (high noise level), Middle left: standard rank-1 SVD reconstruction, middle right: RSVD reconstruction (D and F are 2nd deriv matrices. weight parameters $\lambda = \mu = 1.5$). Bottom: comparison of standard SVD $U(:, 1)$ (red) versus P (blue), and $V(:, 1)$ (red, left) vs. Q (blue, right). The actual $\mathbf{u}$ and $\mathbf{v}$ for noiseless input signal in green.

Self-Labeling of Fully Mediating Representations by Graph Alignment

Martijn Oldenhof[0000−0003−4916−3014], Adam Arany, Yves Moreau, and Jaak Simm

ESAT - STADIUS, KU Leuven, Leuven, 3001, Belgium
{martijn.oldenhof,adam.arany,
yves.moreau,jaak.simm}@esat.kuleuven.be

Abstract. To be able to predict a molecular graph structure (W) given a 2D image of a chemical compound (U) is a challenging problem in machine learning. We are interested to learn $f : U \to W$ where we have a fully mediating representation V such that f factors into $U \to V \to W$. However, observing V requires detailed and expensive labels. We propose **graph aligning** approach that generates rich or detailed labels given normal labels W. In this paper we investigate the scenario of domain adaptation from the source domain where we have access to the expensive labels V to the target domain where only normal labels W are available. Focusing on the problem of predicting chemical compound graphs from 2D images the fully mediating layer is represented using the planar embedding of the chemical graph structure we are predicting. The empirical results show that, using only 4000 data points, we obtain up to 4x improvement of performance after domain adaptation to target domain compared to pretrained model only on the source domain. After domain adaptation, the model is even able to detect atom types that were never observed in the original source domain. Finally, on the Maybridge data set the proposed self-labeling approach reached higher performance than the current state of the art. [1]

1 Introduction

Chemical compounds are often represented by a graph representation of their chemical structure. These graph representations are actually a simplification of the chemical compound as it loses some information about the electronic structure of the molecule. However, in the field of drug discovery this graph representation is often used as valuable input for machine learning pipelines. Examples of formats describing the graph representation of a chemical compounds are SMILES [36] and MOLfile [5]. However, especially in patents but also in scientific literature the chemical compound is only described using an image format. Automatically recognizing the chemical structures on these images is valuable

[1] Code available: https://github.com/biolearning-stadius/chemgrapher-self-rich-labeling .

for machine learning approaches to be able to process these sources of chemical compounds.

Learning to recognize a graph structure from 2D images of chemical compounds seems like a fairly simple task for humans. However, for machine learning models it seems that generalization to new domains of images (e.g. different line width, font face) [21] is not happening naturally. When we humans see an image with a graph structure that we do not recognize completely, we start reasoning and analyzing the part of the graph we are not sure about. We humans automatically align the graph part we recognized on the image with the complete graph including the unrecognized part of the graph. One way to finish our graph prediction is to guess the unknown nodes or edges after which we check for correctness. If the graph prediction was correct we know that this guess was most probably correct and we could try to apply this new knowledge to other images.

To be able to do this reasoning on for example images using graph alignment in machine learning we need a detailed (on pixel level) representation. Therefore we assume a fully mediated model [2] where we are interested to learn $f : U \rightarrow W$ having a fully mediating representation V such that f factors into $U \rightarrow V \rightarrow W$, which is visualized in Figure 1. Thus, in order to predict W from U we first need to pass the fully mediating layer, no side paths are allowed. When a fully mediating representation is used some assumptions [23, 25, 26] are made about the mechanism of the underlying process. This mechanistic prior restricts the space of possible models to all the models that follow the mechanistic assumption. We hypothesize that the use of this richer representation (fully mediating representation) enables for a better generalization. Additionally, as an interesting side effect, we observe that the mechanistic assumption allows for a better interpretability of the underlying model.

In the case of optical graph recognition of chemical compounds from 2D images, the fully mediating layer is represented using the planar embedding of the chemical graph structure we are predicting. In order to learn the planar embedding of a chemical graph structure, we start from a model described in Oldenhof et al. [21] which has two steps: an image segmentation and an image classification step. To train this model, **pixel-wise** annotations are needed for every image describing precise locations of nodes and edges in the graph (planar embedding) which we will call rich or detailed labels in our setup (V). However, these rich labels are not always available and implies a manual process where intermediate organic chemistry knowledge is required. In the more common cases, data sets only contain 2D images of chemical compounds (U in Figure 1) and on the other side the final output in SMILES[36] or MOLfile[5] format (W in Figure 1). These formats describe the graph structure of the chemical compound but not the particular planar embedding of this graph structure (V in Figure 1) in the context of the image. To solve this problem, we propose a **graph aligning** approach that generates rich labels V given normal labels W. This method would enable learning of the fully mediating representations given only normal labels W. In the Figures 7, 8 and 9 in Appendix A.4 examples of U, V and W are shown.

In section 4 we empirically evaluate our domain adaption method. We observe that compared to the non-adapted model we drastically increase accuracy even on atoms and bond that were not present in source domain.

Key contributions: (1) we propose a novel rich labeling framework by introducing the use of a fully mediating layer, (2) in the case of graph recognition we show that the rich labeling can be performed by graph alignment, (3) we show it enables data efficient domain adaption and (4) reaches state-of-the-art performance on Maybridge compound data set.

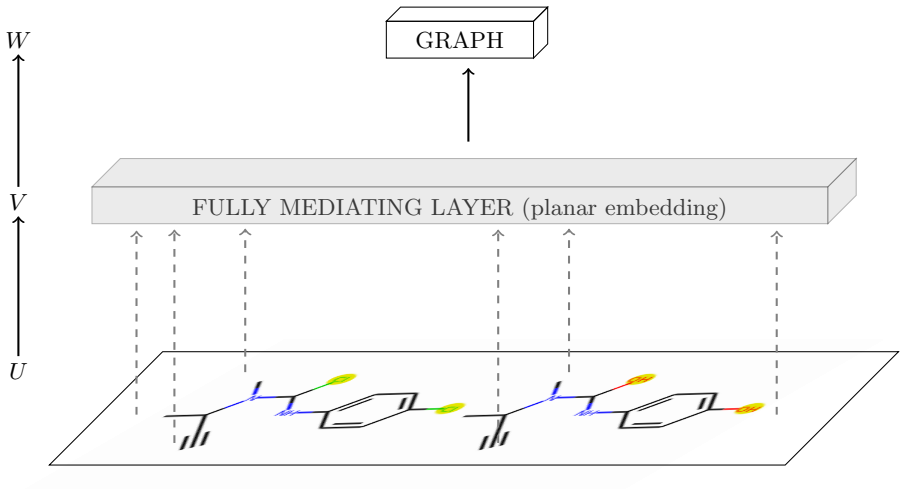


Fig. 1: We are interested to learn $f : U \to W$ having a fully mediating representation V such that f factors into $U \to V \to W$. In the case of optical graph recognition of chemical compounds from 2D images, the fully mediating layer is represented using the planar embedding of the chemical graph structure we are predicting.

2 Related Work

Structural scene representation and visual reasoning. Our work has similarities with research done on structural scene representation and visual reasoning [11, 19, 41]. The disentanglement of the reasoning and the representation described in Yi et al. [41] enables the model to solve complex reasoning tasks. In our work the complex reasoning task would be graph alignment which is disentangled from the optical graph recognition.

Slot Attention. Our method is related with a method called slot attention [17] where the Hungarian algorithm [16] is incorporated in a model for object detection. This Hungarian algorithm is limited to only sets while in our

case we need to map more complicated structures composed of different atoms connected with different bond for which we need graph alignment in order to adapt iteratively a model to a new target domain.

Image to Graph methods. In the field of computational chemistry there are several tools available [7, 18, 20–22, 33, 34] to convert an 2D image of a chemical compound to a SMILES [36] format or similar which in fact represents a graph structure of a chemical compound. Also for road extraction from satellite images there are several methods available [3, 8, 12]

Graph Matching. In computer vision graph alignment is usually known as graph matching. It can be useful to (1) locate objects from features [10], (2) to transfer knowledge [42] and (3) to find matches in database [13]. Also for comparing social networks graph matching can be very important to allow to uncover identities of communities [14]. In chemistry, comparing graphs can be helpful to identify identical chemicals, substructures or maximum common part of chemicals. In the work of Willett et al. [37] an overview is presented about the use of similarity searches in chemical databases.

Domain Adaptation In the work of Kouw and Loog [15] a comprehensive overview is given for domain adaptation methods when labels for the target domain are not available. Our method has some similarities with semi-supervised iterative self-labeling [4, 27] approaches where predictions on a data set of a new domain of a pre-trained model are used as pseudo-labels and used to retrain the model again iteratively until convergence. In the work of Das and Lee [6] even a graph matching loss is first used to learn a domain invariant representation for source and target domain after which the use of pseudo-labels show a significant improvement of performance. In our work the graph matching is used for a different purpose as opposed to the the work of Das and Lee [6]. Graph matching is used in our work to generate rich labels given the 'normal' labels we have from the target domain. This is where our method also differs from other semi-supervised methods for domain adaptation when no target label information at all is assumed and no distinction is made between rich and 'normal' labels.

Weak Supervision In our setup we use the term 'rich' or 'detailed' labels to differentiate from the normal labels. We would like to contrast these 'rich' labels with the term 'strong labels' used in the setting of weak supervision. For example, in the machine learning task of image segmentation pixel-wise labels are needed which are expensive and often not readily available. Therefore, weak supervision methods have been developed to address this issue. Weak supervision can be used to help image segmentation by only using image labels (no pixel-wise labels) [35, 38]. A more general framework was presented in Xu et al. [39] to be able to learn semantic segmentation from a variety of types of weak labels (*e.g.*, image tags, bounding boxes and partial labels). Another approach is to augment the strong labeled data set using weakly labeled data [40]. However, the main difference with all of the methods mentioned above is that our method does not work on weak labels because our end goal is different. The main goal of our machine learning approach is to help to predict 'normal' labels by using rich labels.

Front Door Criterion Our framework exploits fully mediating variables. A variable is called a mediator when it meets several conditions regarding the relationship with other variables as described in Baron and Kenny [2]. Another perspective of the mediating relationship is given by Pearl [23, 25], Pearl et al. [26] who introduce the front door criterion where the mediator actually enables to estimate unbiased causal effects. A more formal interpretation of these causal effects is presented in Pearl [24]. In order to use a mediating model the mediator needs to be identified or assumed first, which is not always straightforward. In our setup (see Figure 1), the assumption means that the relation between input $u \in U$ and planar embedding $v \in V$ is a map and as well as the relation between v and the final graph $w \in W$. Furthermore, we assume no side paths from u to w.

3 Self-Labeling of Fully Mediating Representations

Our goal is to learn $f : U \to W$ assuming a fully mediating representation V such that f factors into $U \to V \to W$. In order to learn the first part of f ($U \to V$) we need labels for V which are expensive in the case of optical graph recognition of chemical compounds from 2D images where V is represented as the planar embedding of the chemical graph structure. Our method tries to address this issue by iteratively updating the model using self-labeled labels for V by graph aligning the graph predictions using the model from previous iteration with the given true graphs (labels W).

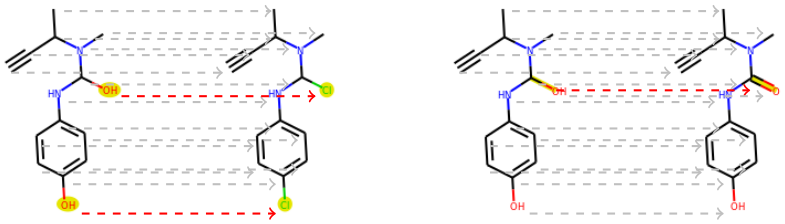
3.1 Graph Alignment

A possible and often used closeness score to compare graphs is the **graph edit distance** [29]: given 2 graphs, not necessarily of equal size and a set of operations, that are $\mathcal{O} = \{\text{vertex/edge/label} \quad \text{insertion/deletion/substitution}\}$, and a cost function $c : \mathcal{O} \mapsto \mathbb{R}$, so we find the cheapest sequence of operations that convert $\mathcal{G}_1$ into $\mathcal{G}_2$, which translates to an optimization problem:

$$\min_{\{e_i\}_{i=1}^{k} \in \mathcal{O}^k : \mathcal{G}_2 = (e_k \circ \ldots \circ e_1) \times \mathcal{G}_1} \sum_{i=1}^{k} c(e_k),$$

Although there are some efficient algorithms available [30–32] in order to compute the graph edit distance, it remains a computational hard problem.

Closely related with the concept of graph edit distance we introduce for our method the map $E(v)$ which gives the allowed operations on a given graph v given a specific constraint. This constraint is a parameter which can be tuned for a specific data set or problem domain. Examples of such constraints are maximum 2 node substitutions or maximum 1 edge substitution as shown in Figure 2.



(a) Example of 2 chemical compounds with graph edit distance of 2 node substitutions.

(b) Example of 2 chemical compound with graph edit distance of 1 edge substitution.

Fig. 2: Two examples of chemical compounds graphs with their graph edit distance. The nodes of the graphs are first aligned before computing the graph edit distance. The node alignments are marked with the gray dashed arrows. The differences after graph alignment are highlighted and the substitutions are marked with the red dashed arrows.

3.2 Method

Let us now say we have a trained neural network model for $f : U \to V$, a projection (not trainable) $\phi : V \to W$, a pair (u, w) of input $u \in U$ and normal label $w \in W$ and we would like to infer rich label $v \in V$ from the given datapoint (u, w). In the setting of chemical structure recognition the projection $\phi : V \to W$ is straightforward (U implies W) and a few examples are shown in Appendix A.4. We also assume the map $E(v)$ which gives all allowed graph edits for the graph v. Let $\hat{v} = f(u)$ be the predicted rich label from the model, then we define a term correcting edit as

Definition 1. *Edit e is a **correcting edit** if when e is applied to the prediction $\hat{v}$ and then projected to the W space the resulting graph is the true graph w (up to isomorphism),* i.e.*,*

$$\phi(e \times \hat{v}) \cong w,$$

where $\times$ is the application of edit to the planar embedded graph $\hat{v}$.

Notice that for a given $\hat{v}$ and w there can be multiple edits that are **correcting edits** which create a dilemma of choosing the best **correcting edit**. Therefore, we make the following assumption:

Assumption 1 *The probability that a correcting edit e results in the true underlying rich label v is monotonely decreasing with respect to the size of edit e (*i.e.*, $|e|$).*

In other words, if we take two correcting edits e_1, e_2 then we assume the following:

$$|e_1| < |e_2| \quad \Rightarrow \quad P(e_1 \times \hat{v} = v) > P(e_2 \times \hat{v} = v)$$

The assumption is based on the fact the *probability of any individual mistake in a graph* by the model is low. This is because if the probability of a mistake would be high the model would not be able to produce a graph with a total of 1-2 edit distance. Thus, the graphs with few edits have low mistake probability and for them the Assumption 1 is valid.

Then we use the following optimisation problem to find the best correcting edit e to convert $\hat{v}$ to rich label v for input u:

$$\mathcal{E}^* = \underset{e \in E(\hat{v})}{\arg\min} |e| \quad \text{such that} \quad \phi(e \times \hat{v}) \cong w,$$

where arg min returns the set of minimal solutions or the empty set if no solutions exist.

There are three possible outcomes of last mentioned optimization problem: (1) no solution is found, (2) a single e is found or (3) multiple equal size e are found. In the optimal case (2) a single e is found so we can label a new v for our given datapoint (u, v). In the case of (1) when no solution is found, no new v is labeled. In the last case (3) when multiple equal size solutions are found there are four options we could do. First (3.1), we could discard the solutions and not label u. Second (3.2), we could take e that results in the highest likelihood for $e \times \hat{v}$ based on the model f. Third (3.3), a solution e is picked uniformly randomly in order to generate the rich label label v. Fourth (3.4), pick e randomly according to the likelihood of $e \times \hat{v}$ in the model f.

This process is repeated for every datapoint (u, w) we have available from the target domain. Thus, several new labels v are found for different datapoints. Once all datapoints are processed these new rich labeled datapoints are added to the training data set after upsampling and our model can be retrained. Upsampling is recommended especially in the case when a low number of normal labelled data points are available compared to the original training dataset. In section 4 different upsampling strategies will be evaluated. After this, a new iteration begins and all available datapoints (u, v) are again processed to find even more

new rich labels v and we can retrain the model again. This iterative process can be repeated until convergence (see Algorithm 1).

Algorithm 1: Iterative algorithm for Self-Labeling of Fully Mediating Representations

Data:
Target domain data $\mathbf{L} = \{(u_i^t, w_i^t)\}_{i=1}^n$
Source domain data $\mathbf{S} = \{(u_j^s, v_j^s)\}_{j=1}^m$ (rich labels)
Result: $f : U \to V$
repeat
  `// Inferring rich labels for target data`
  $\mathbf{T} = []$;
  for (u, w) ***in*** **L do**
    $\hat{v} \leftarrow f(u)$;
    $\mathcal{E}^* \leftarrow \underset{e \in E(\hat{v})}{\arg\min} |e| \quad \text{such that} \quad \phi(e \times \hat{v}) \cong w$;
    if $\mathcal{E}^*$ *is a not empty* **then**
      $e \leftarrow \text{choose}(\mathcal{E}^*)$;
      $v \leftarrow e \times \hat{v}$;
      appendRichLabels($\mathbf{T}, (u, v)$);
    end
  end
  $\mathbf{T} \leftarrow \text{UpSample}(\mathbf{T})$;
  $f \leftarrow \text{RetrainModel}(\mathbf{S}, \mathbf{T})$;
until Converged(f);

4 Experiments

For the experiments we focus on the problem of predicting chemical compound graphs from 2D images where the fully mediating layer is represented using the planar embedding of the chemical graph structure we are predicting. In order to measure empirically the performance of our method of self-labeling fully mediating representations we perform three steps. (1) We pre-train (training details in Appendix A.2) a ChemGrapher [21] model (summarized in Appendix A.1) wherefore, corresponding to the pipeline described in the work of Oldenhof et al. [21], we sample around 130K chemical compounds from ChEMBL [9] in SMILES format and artificially generate, using an RDKit fork [1], a rich labeled dataset with 2D images of chemical compounds. (2) Secondly, we test the baseline performance of this pre-trained model on two different test sets from two different target domains than the source domain of the pre-trained model. (3) Thirdly, we apply our domain adaptation method and measure performance again on the two target domains.

For the first target domain we take a data set from the work from Staker et al. [33], which we will call Indigo data set. For the second target domain we take the data set which was published by the developers of MolRec [18] which we will call the Maybridge data set. Both data sets provide 2D images from a chemical

compound together with corresponding identifier of a the chemical compound like SMILES [36] or MOLfile [5]. These identifiers describe the graph structure of the chemical compound however they do not provide the planar embedding of the graph (e.g. no information about the pixel coordinates of every node or edge in the image). Visually we can also observe that the Maybridge dataset contains images where the style is closer related to the training images style used for the pre-trained model compared with the images in Indigo dataset where the style of images is quite different. Therefore we expect a significant worse starting performance of the pre-trained model on the Indigo dataset compared with the Maybridge dataset.

Dataset	**Orig. Size.**	**# samples to be considered for self-rich-labeling**	**# Test samples**
Indigo	50,000	4,000	1,000
Maybridge	5,740	4,000	1,000

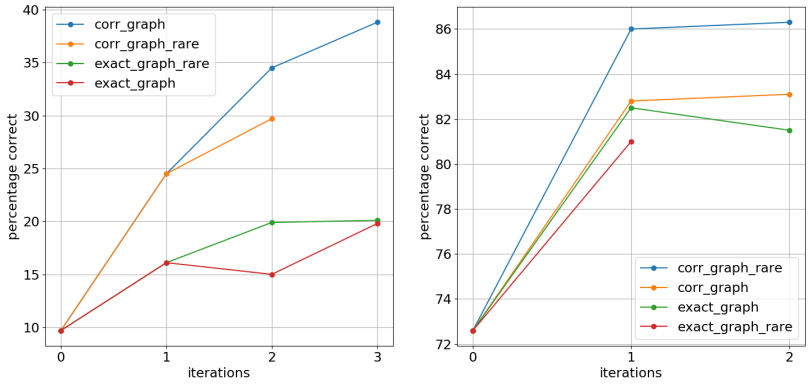
Table 1: Summary of datasets from the 2 different target domains

From both data sets we randomly sample 5,000 datapoints which are split in 4,000 datapoints used for our method and 1,000 datapoints to measure performance on (summarized in Table 1). When processing the 4,000 datapoints our method will be able to generate rich labels for the datapoints where the graph prediction could be graph aligned with the true graph. As the number of rich labeled datapoints this way is maximum 4,000 we will upsample them (x number of copies) before adding them to the training data set. In our experiments we differentiate between two strategies of upsampling. One way is to upsample all the rich labeled data points equally from the target domain to a fixed number, for example 20,000. Another way is to take into account, while upsampling, the number of atom types that are rich labeled and make sure that the rare atom types are upsampled to a specific threshold.

One important tuning parameter in our method is the number of allowed operations. For our experiments we will try two different values for this parameter. Firstly, we set this parameter to zero meaning we do not allow any operation for graph alignment. We will call this **exact graph alignment**. Secondly, we allow a maximum of 2 node substitutions or a maximum of 1 edge substitution for graph alignment, which we will call **correcting graph alignment**.

In total we will measure the performance of 4 variations of our method (varying allowed operations and upsampling strategy) on both data sets. The performance we will measure is the accuracy of $U \rightarrow W$ as we only have access to the normal labels of target domain. However, we assume that if the final graph prediction is correct (W) it is highly likely that also the planar embedding (V) is correct. As our method is an iterative method we will report results for every iteration starting with the initial performance before applying our method.

The results of these experiments are summarized in Figure 3. We observe that all variations of our method are able to improve performance on target domain compared with initial pre-trained model on source domain. On the Indigo data set the best variation is even able to obtain 4x improvement. The best variation of our method on the Indigo data set was using **correcting graph alignment** without upsampling of rare atom types while on the Maybridge data set the best variation was also using **correcting graph alignment** but with upsampling of rare atom types. Some of the underperforming variations of our method were stopped early in order to save computational resources.

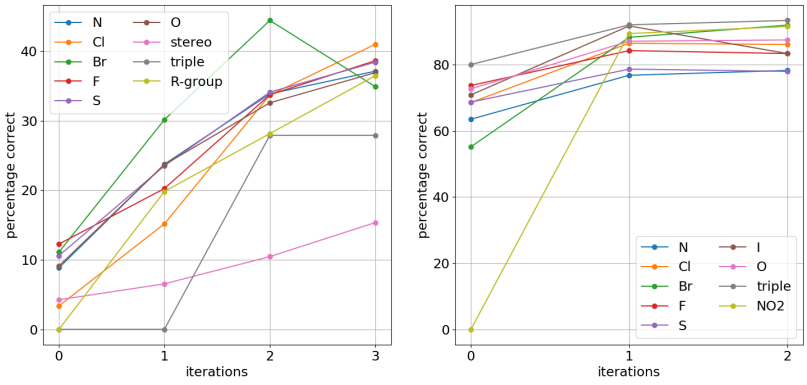


(a) Results on Indigo data set (b) Results on Maybridge data set

Fig. 3: Comparison performance of methods on Indigo and Maybridge data set. Self-labeling by **correcting graph alignment** is clearly better performing than when **exact graph alignment** is used. Sometimes upsampling of rare atoms to a specific threshold (note postfix *_rare*) before retraining of model can boost performance. Performance on target domain at iteration 0 is the performance of pre-trained (on source domain) ChemGrapher before domain adaptation.

We choose the best variation of our method for every data set and analyze the performance on different atom and bond types per iteration. We measure for every atom or bond type the percentage of graphs predicted correctly from the total number of graphs containing that specific atom or bond type per iteration, which is visualized in Figure 4. Most of the performances of the different atom and bond types increase per iteration for both data sets even when initial performance was 0%.

The atom types where initial performance was 0% are atom types never seen before in source domain. For example in the Indigo data set there are compounds with atom labels like R1, R2 and R3 representing R-groups which were not



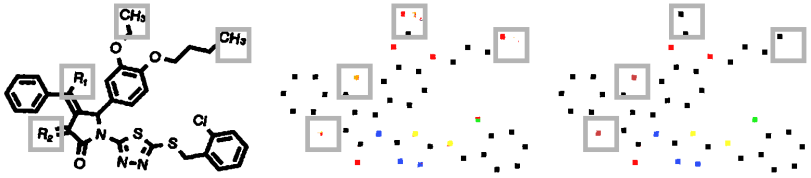
(a) Results on Indigo data set (b) Results on Maybridge dataset

Fig. 4: We take the best performing methods and analyze their performance on different atom and bond types per iteration. We observe that for some atom types the method is able to increase performance even though initial performance was 0%. This is the case in for example R-groups in Indigo data set or superatom NO_2 in Maybridge data set.

present in the original data set from the source domain. For illustration purposes we visualize in Figure 5 the segmentation step which forms part of the graph recognition model used in this study. In the initial segmentation from the pre-trained model we can clearly see that the model confuses the R-group atoms with the oxygen atom type and the hydrogen atom type. After applying our method the model is able to make correct predictions. In the same Figure 5 we also observe that in the Indigo data set carbon sometimes also is represented using a C which was never the case in the original data set. The initial segmentation mainly confuses these carbon atom types with the oxygen atom type. After applying our method the model again makes the correct prediction.

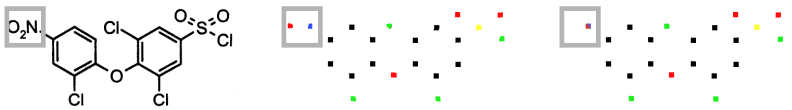
Similarly, the superatom NO_2 present in the Maybridge data set was never observed in the source domain. However, again after applying our method the model is able to detect superatom NO_2 correctly. We illustrate the segmentation step of the graph recognition model in Figure 6 for an example image taken from the Maybridge data set. We observe that in the initial segmentation the pre-trained model confuses NO_2 with nitrogen atom and also oxygen atom which chemically is not the correct prediction. In the final segmentation after applying two iterations of our method the newly trained model is able to make the correct prediction.

Additionally Figure 5 and Figure 6 also show an interesting side effect when using a fully mediating representation. Consider a classical model where input is an image and output is SMILES. When the output prediction of the model



(a) Input image (b) Initial Segmentation (c) Final Segmentation

Fig. 5: Comparison initial segmentation with final segmentation after applying self-labeling of fully mediating representations for Indigo data set. We observe that the initial model is making mistakes on the R-group atom type and carbon represented with a 'C'. In the final model we see that now predictions are all correct.



(a) Input image (b) Initial Segmentation (c) Final Segmentation

Fig. 6: Comparison initial segmentation with final segmentation after applying self-labeling of fully mediating representations for Maybridge data set. The initial model predicts the superatom NO_2 as two separate atoms O and N which is chemically not correct. The final model makes the correct prediction.

is incorrect it is not clear in which part of the image the mistake was made but in the case of having available the planar embedding (mediation representation) the expert can see where and how the mistakes happened. This makes the model more interpretable.

Finally we compare in Table 2 the resulting best performance of the model after applying our method on the Maybridge data set with several other methods available. We observe that our approach enables to reach higher performance than the current state of the art. For the freely available tools OSRA [7] and Molvec [28] we measured the performance using the same randomly 1000 datapoints from the Maybridge dataset. For MolRec [18] this was not possible but we report for information the performance on the total Maybride dataset as reported in the work of M. Sadawi et al. [18]. Finally for ChemGrapher [21] we measured performance using three different training datasets. Firstly, we measure the performance when we only have access to the source domain (generated using RDKit [1]). Secondly, we measure performance using the same training dataset from source domain but adding upsampled (100 copies) 20 handpicked manually rich labeled datapoints from the target Maybridge domain (as was done in the work of Oldenhof et al. [21]). Finally, instead of manually rich labeling datapoints, we process the 4,000 datapoints from Maybridge target domain where our method will be able to generate rich labels for the datapoints where the graph prediction could be graph aligned with the true graph, after which these rich labeled datapoints are added to the training dataset.

Method	**Training Dataset**		**Accuracy**
	Source domain	Target domain	
OSRA (v2.1.0) [7]	N/A	N/A	80.4%
Molvec (v0.9.8) [28]	N/A	N/A	78.4%
ChemGrapher [21]	130K images	N/A	72.6%
ChemGrapher [21] (using manually rich-labeling)	130K images	40 manually handpicked and rich-labeled images (upsampled)	81.6%
Proposed domain adaptation	130K images	4,000 non-rich labeled	**86.3%**
MolRec [18]	N/A	N/A	83.8%[from [18]]

Table 2: Comparison performance on Maybridge data set. We observe that our approach enables to reach higher performance than the current state of the art. Most of the tools available for chemical graph recognition are rule based approaches for which a training dataset is not relevant.

5 Conclusion

Machine learning models often are faced with the problem to not generalize well to a new domain. This is also the case for chemical graph recognition from images. We have shown that fully mediating layers can be exploited in machine learning models to adapt in data efficient way to new domains, without the need of rich expensive labels as they can be generated using our method. In the case of chemical graph recognition we empirically show that our method is able to adapt to a new domain of chemical compounds, with **previously unobserved** atom or bond types. Our rich-labeling method required only 4,000 normal labeled points in the target domain to go from 10% accuracy to 39%, i.e., almost 4x improvement in the difficult Indigo data set. Using more normal labeled points and more iterations would most probably give a higher resulting accuracy. Furthermore, on Maybridge data set, again using only 4,000 images, we reached high accuracy obtaining better performance than the current state of the art.

Effective tools of chemical structure recognition from images enable access to the knowledge in chemical literature which is currently only available through expensive chemistry databases. We believe it as an important step towards open pharmaceutical science.

It would be interesting to apply this method to other contexts where the output of a machine learning model could be represented with a graph structure. For example, the case of structural scene representation, where a scene could be represented using a graph where every vertex could represent an object and every edge would represent the relations between the objects (e.g. side-by-side, on-top-of, under). This structural scene could be in form of 2D images or it could be even generalized to 3D models, where point clouds are available and one is interested to transform them into 3D graphs of connected parts.

Acknowledgments MO, AA, YM, and JS are funded by (1) Research Council KU Leuven: C14/18/092 SymBioSys3; CELSA-HIDUCTION, (2) Innovative Medicines Initiative: MELLODDY, (3) Flemish Government (ELIXIR Belgium, IWT: PhD grants, FWO 06260) and (4) Impulsfonds AI: VR 2019 2203 DOC.0318/1QUATER Kenniscentrum Data en Maatschappij. Computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government – department EWI. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] Fork of the official sources for the rdkit library (2020), URL `https://github.com/biolearning-stadius/rdkit`, accessed on 12.11.2020

[2] Baron, R.M., Kenny, D.A.: The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. Journal of personality and social psychology **51**(6), 1173 (1986)
[3] Belli, D., Kipf, T.: Image-conditioned graph generation for road network extraction. arXiv preprint arXiv:1910.14388 (2019)
[4] Bruzzone, L., Marconcini, M.: Domain adaptation problems: A dasvm classification technique and a circular validation strategy. IEEE transactions on pattern analysis and machine intelligence **32**(5), 770–787 (2009)
[5] Dalby, A., Nourse, J.G., Hounshell, W.D., Gushurst, A.K.I., Grier, D.L., Leland, B.A., Laufer, J.: Description of several chemical structure file formats used by computer programs developed at molecular design limited. J. Chem. Inf. Comput. Sci. **32**(3), 244–255 (1992), URL `https://pubs.acs.org/doi/abs/10.1021/ci00007a012`
[6] Das, D., Lee, C.G.: Graph matching and pseudo-label guided deep unsupervised domain adaptation. In: International conference on artificial neural networks, pp. 342–352, Springer (2018)
[7] Filippov, I.V., Nicklaus, M.C.: Optical structure recognition software to recover chemical information: Osra, an open source solution. J. Chem. Inf. Model. **49**(3), 740–743 (2009), URL `https://doi.org/10.1021/ci800067r`
[8] Gao, L., Song, W., Dai, J., Chen, Y.: Road extraction from high-resolution remote sensing imagery using refined deep residual convolutional neural network. Remote Sensing **11**(5), 552 (2019)
[9] Gaulton, A., Hersey, A., Nowotka, M., Bento, A.P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L.J., Cibrián-Uhalte, E., et al.: The chembl database in 2017. Nucleic acids research **45**(D1), D945–D954 (2017)
[10] Gold, S., Rangarajan, A.: Graduated assignment graph matching. In: Proceedings of International Conference on Neural Networks (ICNN'96), vol. 3, pp. 1474–1479, IEEE (1996)
[11] Han, C., Mao, J., Gan, C., Tenenbaum, J., Wu, J.: Visual concept-metaconcept learning. In: Advances in Neural Information Processing Systems, pp. 5001–5012 (2019)
[12] Henry, C., Azimi, S.M., Merkle, N.: Road segmentation in sar satellite images with deep fully convolutional neural networks. IEEE Geoscience and Remote Sensing Letters **15**(12), 1867–1871 (2018)
[13] Kisku, D.R., Rattani, A., Grosso, E., Tistarelli, M.: Face identification by sift-based complete graph topology. In: 2007 IEEE workshop on automatic identification advanced technologies, pp. 63–68, IEEE (2007)
[14] Kong, X., Zhang, J., Yu, P.S.: Inferring anchor links across multiple heterogeneous social networks. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pp. 179–188 (2013)
[15] Kouw, W.M., Loog, M.: A review of domain adaptation without target labels. IEEE transactions on pattern analysis and machine intelligence (2019)
[16] Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)

[17] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. arXiv preprint arXiv:2006.15055 (2020)

[18] M. Sadawi, N., Sexton, A., Sorge, V.: Chemical structure recognition: A rule based approach. Proc. SPIE **8297**, 32– (01 2012), URL `https://doi.org/10.1117/12.912185`

[19] Mao, J., Gan, C., Kohli, P., Tenenbaum, J.B., Wu, J.: The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. arXiv preprint arXiv:1904.12584 (2019)

[20] McDaniel, J.R., Balmuth, J.R.: Kekule: Ocr-optical chemical (structure) recognition. J. Chem. Inf. Comput. Sci. **32**(4), 373–378 (1992), URL `https://doi.org/10.1021/ci00008a018`

[21] Oldenhof, M., Arany, A., Moreau, Y., Simm, J.: Chemgrapher: optical graph recognition of chemical compounds by deep learning. Journal of Chemical Information and Modeling **60**(10), 4506–4517 (2020)

[22] Park, J., Rosania, G., A Shedden, K., Nguyen, M., Lyu, N., Saitou, K.: Automated extraction of chemical structure information from digital raster images. Chem. Cent. J. **3**, 4 (03 2009), URL `https://doi.org/10.1186/1752-153X-3-4`

[23] Pearl, J.: Causal diagrams for empirical research. Biometrika **82**(4), 669–688 (1995)

[24] Pearl, J.: Direct and indirect effects. In: Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence, pp. 411–420 (2001)

[25] Pearl, J.: Causality. Cambridge university press (2009)

[26] Pearl, J., et al.: Models, reasoning and inference. Cambridge, UK: CambridgeUniversityPress (2000)

[27] Pérez, Ó., Sánchez-Montañés, M.: A new learning strategy for classification problems with different training and test distributions. In: International Work-Conference on Artificial Neural Networks, pp. 178–185, Springer (2007)

[28] Peryea, T., Katzel, D., Zhao, T., Southall, N., Nguyen, D.T.: Molvec: Open source library for chemical structure recognition. In: ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY, vol. 258, AMER CHEMICAL SOC 1155 16TH ST, NW, WASHINGTON, DC 20036 USA (2019)

[29] Sanfeliu, A., Fu, K.S.: A distance measure between attributed relational graphs for pattern recognition. IEEE transactions on systems, man, and cybernetics (3), 353–362 (1983)

[30] Serratosa, F.: Fast computation of bipartite graph matching. Pattern Recognition Letters **45**, 244–250 (2014)

[31] Serratosa, F.: Computation of graph edit distance: reasoning about optimality and speed-up. Image and Vision Computing **40**, 38–48 (2015)

[32] Serratosa, F.: Speeding up fast bipartite graph matching through a new cost matrix. International Journal of Pattern Recognition and Artificial Intelligence **29**(02), 1550010 (2015)

[33] Staker, J., Marshall, K., Abel, R., McQuaw, C.M.: Molecular Structure Extraction from Documents Using Deep Learning. J. Chem. Inf. Model. **59**(3), 1017–1029 (Mar 2019), ISSN 1549-9596, https://doi.org/10.1021/acs.jcim.8b00669, URL `https://doi.org/10.1021/acs.jcim.8b00669`

[34] Valko, A.T., Johnson, A.P.: Clide pro: The latest generation of clide, a tool for optical chemical structure recognition. J. Chem. Inf. Model. **49**(4), 780–787 (2009), URL `https://doi.org/10.1021/ci800449t`

[35] Vezhnevets, A., Ferrari, V., Buhmann, J.M.: Weakly supervised structured output learning for semantic segmentation. In: 2012 IEEE conference on computer vision and pattern recognition, pp. 845–852, IEEE (2012)

[36] Weininger, D.: SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. **28**(1), 31–36 (1988), URL `https://pubs.acs.org/doi/abs/10.1021/ci00057a005`

[37] Willett, P., Barnard, J.M., Downs, G.M.: Chemical similarity searching. . Chem. Inf. Comput. Sci. **38**(6), 983–996 (1998)

[38] Xu, J., Schwing, A.G., Urtasun, R.: Tell me what you see and i will show you where it is. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3190–3197 (2014)

[39] Xu, J., Schwing, A.G., Urtasun, R.: Learning to segment under various forms of weak supervision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3781–3790 (2015)

[40] Xu, Z., Huang, S., Zhang, Y., Tao, D.: Augmenting strong supervision using web data for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision, pp. 2524–2532 (2015)

[41] Yi, K., Wu, J., Gan, C., Torralba, A., Kohli, P., Tenenbaum, J.: Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In: Advances in neural information processing systems, pp. 1031–1042 (2018)

[42] Zhang, H., Xiao, J., Quan, L.: Supervised label transfer for semantic segmentation of street scenes. In: European Conference on Computer Vision, pp. 561–574, Springer (2010)

A Appendix

A.1 Architecture Summary of Graph Recognition tool

Every iteration of our method we need to train the graph recognition tool described in Oldenhof et al. [21]. This graph recognition tool is built using a combination of different convolutional neural networks. The first part is a semantic segmentation network to pixel-wise predict every atom, bond and charge type. The second part consists of three classification networks to classify every segment predicted by the semantic segmentation network. After the first step of the ChemGrapher model [21], the segmentation network, the predicted segments are

processed so that for every segment the center of mass is calculated. These centers of mass would be the atom/bond/charge candidates to be classified by the classification networks.

Table 3: Summary of the layers of the segmentation network

Layer	Kernel	Nonlinearity	Padding	Dilation
conv1	3x3	ReLU	1	no dilation
conv2	3x3	ReLU	2	2
conv3	3x3	ReLU	4	4
conv4	3x3	ReLU	8	8
conv5	3x3	ReLU	8	8
conv6	3x3	ReLU	4	4
conv7	3x3	ReLU	2	2
conv8	3x3	ReLU	1	no dilation
last	1x1	none	no padding	no dilation

Table 4: Different layers in the classification network

Layer	Kernel	Nonlinearity	Padding	Dilation
depthconv1	3x3	ReLU	1	no dilation
conv2	3x3	ReLU	2	2
conv3	3x3	ReLU	4	4
conv4	3x3	ReLU	8	8
conv5	3x3	ReLU	1	no dilation
global maxpool	input size	None	no padding	no dilation
last	1x1	None	no padding	no dilation

A.2 Training details for graph recognition tool

Training details of the graph recognition tool for every iteration of our method are summarized in Table 5. The input images used for training of the different networks are a mix if images from source domain and upsampled rich labeled images from target domain. For pretraining of the ChemGrapher model only images from source domain were used. The training was performed using a compute node with 2 NVIDIA v100 GPUs with 32GB of memory.

Table 5: Training details for different networks

Network	#input images source domain	#input images target domain (upsampled)	#epochs	walltime	minibatch size	learning rate
Segm. network	114K	20K	5	24h	8	0.001
Atom Clas.	12.4K	2.6K	2	8h	16	0.001
Charge Clas.	12.4K	2.6K	2	8h	16	0.001
Bond Clas.	4.4K	2.1K	2	4h	64	0.001

A.3 Computational cost per rich-labeling iteration

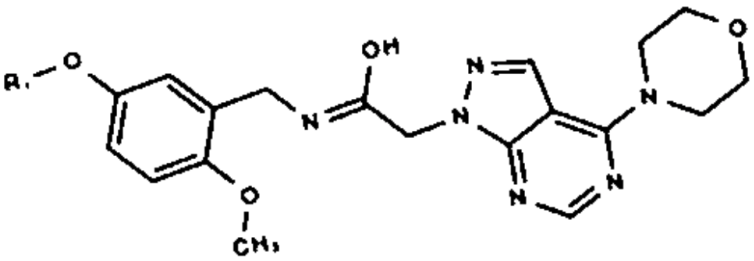
In the following Table 6 the computational cost for 1 rich-labeling iteration is summarized including all steps: (re)training, predicting and graph aligning rich-labeling.

Table 6: Computational costs per rich-labeling iteration

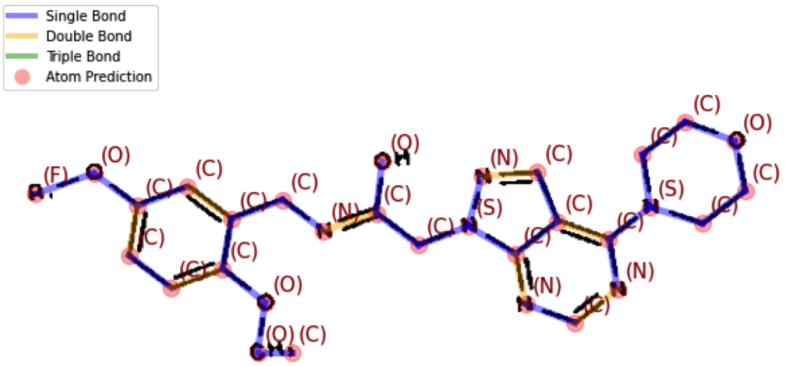
	Training	Predict	Graph Aligning	
Hardware	2 NVIDIA v100 GPUs	1 NVIDIA v100 GPU	Intel Xeon Gold 6240 2.6Ghz	
Dataset	Source+Target domain	Indigo/Maybride	Indigo	Maybridge
#datapoints	see Table 5	4,000	4,000	4,000
Walltime	∼44h (details Table 5)	∼2h	∼40min	∼3min

A.4 Examples of cases where graph alignment fails

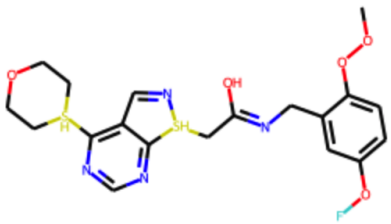
We would like to showcase some examples where the constrained (max 2 node substitutions or max 1 edge substitution) graph alignment fails. At the same time it is important to note that our proposed domain adaptation method is an iterative method, so if a graph alignment fails in a previous iteration it could succeed in a next one when the new model makes a new graph prediction closer to the true graph.



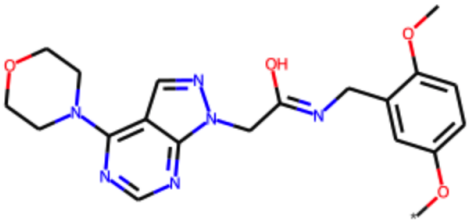
(a) Input Image(U)



(b) Planar embedding prediction (V')

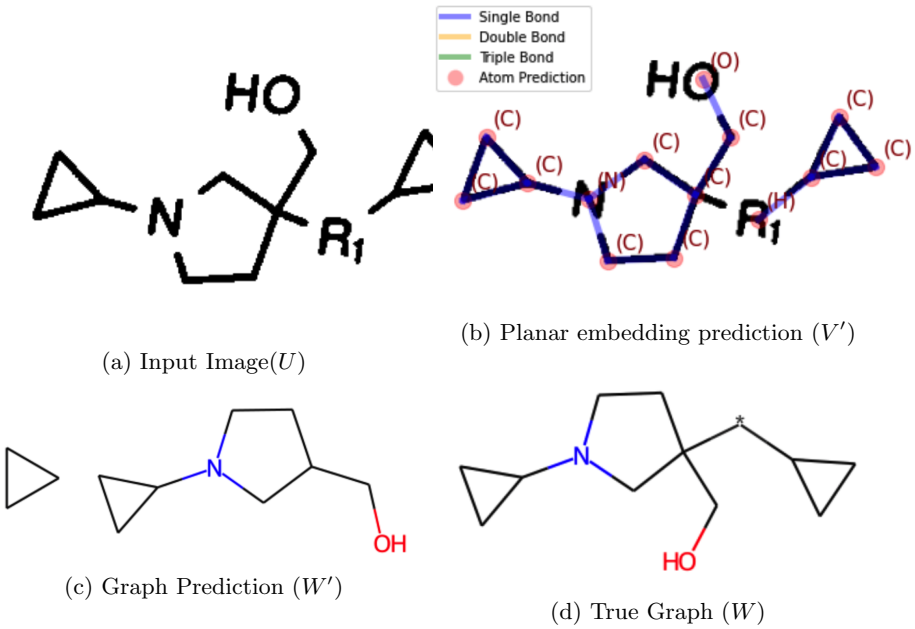


(c) Graph Prediction (W')



(d) True Graph (W)

Fig. 7: Example 1: It is clear that to align the graph prediction W' with the true graph W more than 2 node substitutions are needed. So no rich labeling is possible for this example in this iteration.



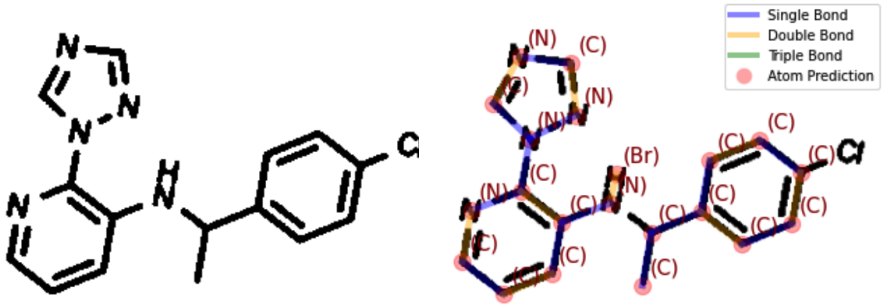
(a) Input Image(U)

(b) Planar embedding prediction (V')

(c) Graph Prediction (W')

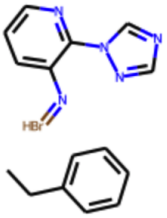
(d) True Graph (W)

Fig. 8: Example 2: It is clear that alignment of the graph prediction W' with the true graph W can not be solved with only substitutions.

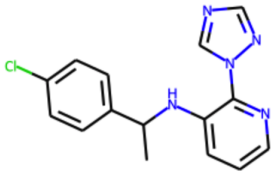


(a) Input Image(U)

(b) Planar embedding prediction (V')



(c) Graph Prediction (W')



(d) True Graph (W)

Fig. 9: Example 3: It is clear that alignment of the graph prediction W' with the true graph W can not be solved with only substitutions.

ExClus: Explainable Clustering on Low-dimensional Data Representations*

Xander Vankwikelberge, Bo Kang, Edith Heiter, and Jefrey Lijffijt

Ghent University, Ghent, Belgium

Abstract. Dimensionality reduction and clustering techniques are frequently used to analyze complex data sets, but their results are often not easy to interpret. We consider how to support users in interpreting apparent cluster structure on scatter plots where the axes are not directly interpretable, such as when the data is projected onto a two-dimensional space using a dimensionality-reduction method. Specifically, we propose a new method to compute an interpretable clustering automatically, where the explanation is in the original high-dimensional space and the clustering is coherent in the low-dimensional projection. It provides a tunable balance between the complexity and the amount of information provided, through the use of information theory. We study the computational complexity of this problem and introduce restrictions on the search space of solutions to arrive at an efficient, tunable, greedy optimization algorithm. This algorithm is furthermore implemented in an interactive tool called ExClus. Experiments on several data sets highlight that ExClus can provide informative and easy-to-understand patterns, and they expose where the algorithm is efficient and where there is room for improvement considering tunability and scalability.

Keywords: Dimensionality reduction · clustering · explainable AI · exploratory data analysis · hierarchical clustering · t-SNE.

1 Introduction

Artificial intelligence methods exceed human performance on many tasks and thus have found widespread use, yet the resulting models are often black boxes. There is a growing demand to create scalable human-friendly implementations. The creation of explainable white-box artificial intelligence methods is necessary to have users trust, manage, and use these in making crucial decisions [1, 8].

This need is also present in the area of clustering and dimensionality reduction methods for high-dimensional data. Clustering and dimensionality reduction methods are frequently employed to get a grasp of the high-level structure

* The research leading to these results has received funding from the ERC under the EU's Seventh Framework Programme (FP7/2007-2013) (ERC Grant Agreement no. 615517) and under the EU's Horizon 2020 research and innovation programme (ERC Grant Agreement no. 963924), from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, and from the FWO (project no. G091017N, G0F9816N, 3G042220)

of data. Especially non-linear dimensionality reduction methods such as Isomap [16], LLE [12], t-SNE [10], and UMAP [11] manage to effectively map data onto a low-dimensional space, retaining the relative distances from the high-dimensional space, but this low-dimensional space is often difficult to understand [14].

Related work. Possible solutions in several directions have been studied: we may explore the 2D projection by letting the data glyphs correspond to specific attributes (e.g., color the points using the values for a specific attribute) or draw attribute isolines (as in DimReader [7]). Such solutions are limited to a single or very few attributes at a time and it is not obvious how to make them practical for data whose original dimensionality is large (e.g., from 20 attributes or more).

There also exist approaches that focus on data points rather than attributes: 'Forward projections' show how points would move in the low-dimensional embedding if their attributes would change, and we may also do the inverse exercise of 'backward projection', i.e., how the attributes would need to change to move a point in a certain direction (which does not have a unique solution, so an inductive bias is necessary to resolve this) [3]. However, it is not practical to learn the structure of a large data set by exploring each point individually. 'Probing clusters' [15] means to explore the feature values for manually selected sets of points that for example appear to form a cluster. This may lead to insights on the high-level structure. Yet, for high-dimensional data we are still left with the problem that this does not scale to a large number of attributes.

Most similar to our work is the Clustrophile 2 tool [4], which lets users explore a diverse set of pre-generated clusterings by means of a scatter plot showing a low-dimensional projection, as well as an attribute similarity matrix and/or a decision tree for the cluster assignment. However, the decision tree does not explicitly show how each cluster stands out, while the similarity matrix is only a visual aid that puts the burden of the comparison fully on the user, who are likely quickly overwhelmed for data with a large number of attributes.

Contributions. We propose in this paper to take a very different approach: to compute an interpretable clustering that facilitates the analysis of a scatter plot of dimensionality-reduced data. We do this in several steps: (1) first we use information theory to quantify the informativeness of the mean and variance statistics for a given subset of points and given subset of attributes. (2) To control the trade-off between the complexity of the clustering and the amount of information gained, we introduce a simple notion of complexity for such a 'bicluster pattern', which gives us control of how complicated the explanations for the clusters may be. (3) Integrating this with an approach to cluster the data on the low-dimensional representation, we obtain coherent clusters on the dimensionality-reduced scatter plot along with a subset of attributes that are informative (in the information-theoric sense), to explain this cluster. (4) Using this, we attempt to automatically generate an optimal explainable clustering on low-dimensional representations of data sets, so that users can meaningfully explore complex data with a limited time cost for interpreting apparent structure.

The contributions of this paper are as follows:

– We define bicluster patterns, their informativeness using information theory and a simple notion of their interpretational complexity.
– We derive an algorithm to optimally cluster a dimensionality-reduced scatter plot such that the clusters are as informative as possible using a few attributes in the high-dimensional space.
– We study the computational complexity of this problem and argue for the use of hierarchical clustering to reduce the search space of possible solutions.
– We provide an implementation of that algorithm as well as a browser-based tool called ExClus (see Fig. 1) that can be used to explore data in practice.
– We present experimental results on a few datasets to explore the usefulness of the tool as well as the effect of the two hyperparameters that govern the interpretional complexity of bicluster patterns. We present some early feedback from users and empirically study the scalability of the method.
– We find that ExClus has great potential to further facilitate data exploration with dimensionality-reduction methods and that the method is sufficiently scalable to use also on large data.

The ExClus tool is freely available at https://github.com/aida-ugent/ExClus

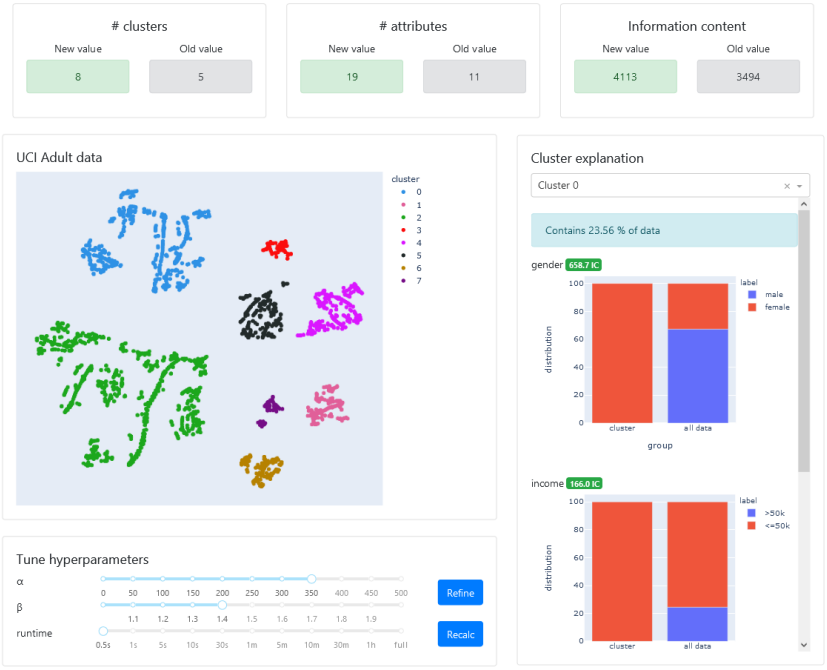


Fig. 1: The ExClus user application. Results are generated by applying the algorithm on the UCI Adult data set, an extract from the 1994 USA census [6].

The paper is structured as follows: in Section 2 we first formalize the type of patterns that we aim to extract and define the informativeness and descriptional complexity of these patterns. We then introduce the optimization algorithm to extract the patterns. In Section 4 we introduce ExClus and the user interface. Experiments are given in Section 5 and conclusions are presented in Section 6.

2 Method

The general idea is to define the substructure that we aim to find as a pattern, which we call a *bicluster pattern*. Informally, this is a subset of points and the mean and variance statistics for a subset of the attributes. The approach then builds upon the framework for subjective interestingness for patterns [5], which suggests the use of information theory to quantify how much information a user gets per time unit spent. There are three key concepts in this framework: information content, description complexity, and subjective interestingness.

The *information content* expresses how much the user learns by showing them a specific pattern. The *description complexity* aims to express the difficulty of understanding the pattern, and the *subjective interestingness* is simply the ratio of the two. The term 'subjective' explicates that how much we learn can only be specified with respect to prior expectations over the data. In practice we will use a prior based on the data, but it may also be chosen subjectively. Barring issues of feasibility, the prior could reflect the actual knowledge of a user.

We discuss each of these concepts in more detail: In Section 2.1, we express the *information content* of a bicluster pattern, i.e., the number of bits of information that we learn by showing the statistics for this bicluster, in comparison to given prior expectations. In Section 2.2, we also introduce the *description complexity* that aims to capture how difficult or time consuming it is to process the presented information, for a human end-user. In Section 2.3, we then formalize the *explainable clustering problem*: to find a set of bicluster patterns that partition the data into clusters that are coherent with respect to a given 2D projection, such that the *subjective interestingness* of the clustering is maximized.

2.1 Bicluster patterns and their information content

Notation. Let $\mathbb{X}$ be an $n \times m$ data matrix with $\mathbb{X}_i$ denoting data point $i \in \{1, \ldots, n\}$ and $\mathbb{X}_{ij}$ the value for the j-th attribute ($j \in \{1, \ldots, m\}$) of $\mathbb{X}_i$. We write $t(j) \in \{\text{bool}, \text{real}\}$ to denote the type of attribute j.

Definition 1. *A bicluster pattern $\mathcal{P}$ is a tuple $(D, A, \mathcal{S})$ with $D \subseteq \{1, \ldots, n\}$ a set of data points indices, $A \subseteq \{1, \ldots, m\}$ a set of attribute indices, and the statistics $\mathcal{S} = \{\mathcal{S}_{A_1}, \ldots, \mathcal{S}_{A_{|A|}}\}$, corresponding to the attributes whose indices are in A. For boolean attributes ($t(j) = bool$), $\mathcal{S}_j \in [0, 1]$ is a frequency and for real-valued attributes ($t(j) = real$) both a mean and standard deviation: $\mathcal{S}_j \in \mathbb{R} \times \mathbb{R}^+$.*

To express how much we learn by observing the statistics for a set of attributes for a subset of the data, we first need to express what we are comparing

against, i.e., we have to define a prior distribution for the data. We take a simple approach here and use as prior the maximum likelihood statistics fitted on the full data, without accounting for any co-variate structure. That is, each attribute is assumed to be independent. Although in some context it may be preferable to already account for co-variates between the attributes, in many cases the independence assumption is good as it is also transparent for the user.

Similarly, the information that we obtain by observing a bicluster pattern are the maximum likelihood statistics $\mathcal{S}$ for the set of points D and the attributes A. The statistics may be used to derive a Maximum Entropy model for the data. As the statistics are assumed independent, indeed also the Maximum Entropy model is independent and we may write it for each attribute separately. It depends on the attribute type, so for brevity we write this model as $\mathcal{M}_{t(j)}(\mathcal{S}_j)$. The Maximum Entropy model is a Gaussian distribution for real-valued attributes and a Bernoulli distribution for boolean attributes.

The information content of a pattern $\mathcal{P}$ is equivalent to the Kullback-Leibler divergence between the prior expectations and the statistics contained in $\mathcal{P}$:

$$I(\mathcal{P}) = \sum_{i=1}^{|D|} \sum_{j=1}^{|A|} D_{KL}\left(\mathcal{M}_{t(A_j)}(\mathcal{S}^D_{A_j}) || \mathcal{M}_{t(A_j)}(\mathcal{S}^{\mathbb{X}}_{A_j})\right). \tag{1}$$

Note that we overloaded D_{KL} here to refer to the KL divergence, but in all other occurrences D indeed refers to a subset of the data points included in a bicluster pattern. The KL divergences are straightforward to compute analytically for both the Bernoulli and Gaussian distribution.

Note that it may happen that the variance of a bicluster for a specific real-valued attribute is zero, in which case the KL divergence will be infinite. Therefore, we add a small value ϵ to all variance estimates. Resolving this in a more robust manner by for example considering the precision of the real-valued numbers is left for future work.

2.2 Description complexity

The aim of the description complexity is to quantify how difficult it is to process the presented information, i.e., how time consuming it is to internalize. Unfortunately we have no realistic models of human cognition so we will just need to work on assumptions. It is important to realize that our aim here is not to do model selection in the statistical sense and we are not presenting the patterns to another computer. The aim is simply to have a formula that is suitably parameterized such that we can balance the amount of information and the complexity of the identified patterns.

In previous papers on subjective interestingness, the description complexity was quantified as a linear function over the number of statistics that is presented to the user. Often, this leads to a more tractable optimization problem. We instead choose to make it explicit that providing more statistics realistically has a superlinear effect on the amount of time to process the information. Because

we are going to present the user not a single bicluster pattern, but a set that partitions the data, we define the description complexity directly for a set of bicluster patterns as α plus the total number of statistics to the power β:

$$C(\{\mathcal{P}_1, \ldots, \mathcal{P}_k\}) = \alpha + \left(\sum_{i=1}^{k} \sum_{j=1}^{|A^{\mathcal{P}_i}|} |\mathcal{S}_{A_j^{\mathcal{P}_i}}| \right)^{\beta}. \tag{2}$$

Here $|\mathcal{S}_{A_j}| = 1$ for boolean attributes and $|\mathcal{S}_{A_j}| = 2$ for real-valued attributes. This formula also includes two hyperparameters, α, and β, which allow for tuning by users and enable the identification of more intuitive solutions.

2.3 Explainable clustering problem

Finally, we want to find a clustering that makes a trade-off between the total information content and description complexity. Hence, we define the subjective interestingness for a set of bicluster patterns as

$$S(\{\mathcal{P}_1, \ldots, \mathcal{P}_k\}) = \frac{\sum_{i=1}^{k} I(\mathcal{P}_i)}{C(\{\mathcal{P}_1, \ldots, \mathcal{P}_k\})}. \tag{3}$$

The missing component is that we have not related these concepts yet to the low-dimensional projection that we started out from. Let $\mathbb{Y}$ denote this low-dimensional—typically 2D—projection of $\mathbb{X}$. We may now define the explainable clustering problem.

Problem 1. The explainable clustering problem is to find a set of bicluster patterns $\{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ ($k \geq 1$) that partition the data (i.e., they cover all data points $\bigcup_{i=1}^{k} D_i = \{1, \ldots, n\}$ and no data point is covered twice $D_i \cap D_j = \emptyset \; \forall i \neq j$) in a coherent manner with respect to $\mathbb{Y}$ and that maximize the subjective interestingness $S(\{\mathcal{P}_1, \ldots, \mathcal{P}_k\})$.

Here we have not defined exactly when a partitioning may be called coherent with respect to $\mathbb{Y}$. Indeed, this may differ per usage scenario and we argue that it is not obvious there may exist a universally applicable definition. We will argue for a particular choice in the following section that also benefits the design of an efficient optimization algorithm.

3 Search algorithm

We are faced with a difficult optimization problem: we want to cluster the data in a given low-dimensional projection, in such a manner that the attribute values are coherent and we are concurrently selecting attributes to explain the clusters.

We first considered a two-step approach; first clustering the data and then computing the optimal explanations. Alternatively, we could derive an iterative optimization scheme, e.g., an EM scheme switching between data point assignment and the explanation attributes. However, we opt for a third solution where we constrain the search space to make it feasible to optimize the cluster assignment, number of clusters, and the attribute selection in an integrated manner.

3.1 The ExClus algorithm

Step 1. We start by computing a hierarchical agglomerative clustering using the Euclidean distance on the projection $\mathbb{Y}$. A tested and popular method for clustering data, this conveniently ensures coherence of the final clustering solution and it constrains the search space. The resulting dendrogram will be used to guide the full clustering, but we will not use the distances in $\mathbb{Y}$ to select clusters.
Step 2. Given the optimization problem, we have to consider how to cut the dendrogram to obtain the best set of bicluster patterns. We are still faced with a large number of possible clusterings. If the tree is sufficiently large and balanced, the number of possible clusterings with k clusters is given the corresponding Catalan number, which is already 1430 for $k = 8$, for example.

To obtain an approximate solution, we optimize the clustering by splitting one cluster of at a time. That is, we start with everything in one cluster, then consider all possible splits for $k = 2$. This can be done in time linear in the size of the dendrogram, as indeed we may exactly split any branch off. For each possible split, we can optimize the attributes by ranking them and greedily considering whether we should add a boolean or a real-valued attribute (because they weigh differently in the description complexity). We obtain indeed the optimal solution for $k = 2$ given the constraints, but then we take the $k = 2$ solution and split this once more in any position in the tree to obtain a solution for $k = 3$, which is not guaranteed to be optimal. The procedure continues for $k > 3$.

This procedure could go on until the number of clusters is equal to the number of samples. This would be time consuming and it is highly unlikely that a solution with very large k would maximize the subjective interestingness. Hence, we implement a time limit to ensure the runtime stays in an acceptable range. This limit can be any value, from a few milliseconds to only stopping when all calculations end.

3.2 Solution refinement

It is not obvious how to tune the hyperparameters, so we expect this to be done through trial-and-error. From experiments presented in Section 5, we find they can have a dramatic effect on the resulting solution and the differences may also be abrupt. Restarting the algorithm from scratch is then not always desirable to find a good solution. Hence, we also introduce a procedure to modify the current solution given new hyperparameter values.

This modification again uses the dendrogram, but instead of starting at one cluster, it starts from the previously obtained result. The procedure then evaluates both whether splitting more clusters off, as described in Section 3.1, or merging some clusters back together that were previously split off results in a higher subjective interestingness.

4 User application

In this section we present the user interface of ExClus, which is based on the previously introduced formalization and algorithm. The application allows test-

ing and intuitive usage of the presented algorithm. In the following, we highlight different aspects of the interface and summarize user feedback.

4.1 Design choices

The interactive tool helps users to understand data sets by generating clusters and their explanations simultaneously. Users can browse through through the explanations at their own pace and tune the algorithm to retrieve more insights. The application is created with Dash, a Python framework for building data science web applications. Figure 1 shows the user interface.

Dashboard. The dashboard, situated at the top of the application, shows some essential information on the currently presented clustering. After tuning the hyperparameters to get different results, it also shows how these values changed. Alone, these values do not tell much, but after tuning, it is helpful the compare how they changed.

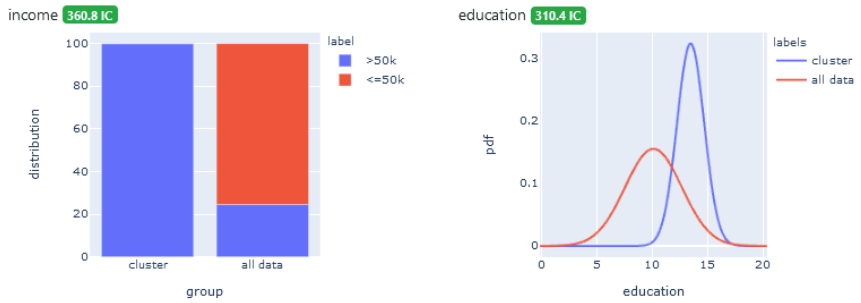
- **# clusters:** How many clusters the algorithm generated.
- **# attributes:** Each cluster needs a certain number of attributes as an explanation (at least one). For example, cluster one might need only one attribute, but cluster two might need four. This number is the sum of all these attributes for all clusters.
- **Information content:** This is the entire clustering's Information Content[1].

Data Embedding. In the top left of the interface we show a scatter plot of the low-dimensional data and display the clustering result from ExClus with different colors. Going from multiple dimensions to only two with a non-linear transformation, which dimensionality reduction techniques as t-SNE [10] do, results in difficult to interpret axes, so the only thing deductible from the graph, besides the clustering, is that points close together on the 2D plot are similar in the high-dimensional space.

Cluster explanations. This part of the application provides the identified explanations of why specific points form a cluster. When users select one cluster using the drop-down menu, we show the relative size of the cluster and the description of attributes that make this cluster distinct from the rest of the data. We sort the attributes by decreasing information content and use different visualizations for binary and real-valued attributes as shown in Figure 2.

For binary attributes, we illustrate the distribution of values in a stacked bar chart. The left bar shows the distribution within the selected cluster and the right bar the distribution of the entire data set. For real-valued attributes, we derived the information content by fitting Gaussian distributions using mean and

[1] It is a deliberate choice to show the Information Content and not the Subjective Interestingness score, as the latter depends on the values of the hyperparameters. It is not sensible to compare those values across different hyperparameter choices, while the Information Content can indeed be compared.



(a) Explanation of a binary attribute. (b) Explanation of a real-valued attribute

Fig. 2: Part of the explanation for cluster four in Figure 1.

standard deviation as parameters. To compare the selected cluster to the entire dataset, we plot both probability density functions for the Gaussian distributions in the prior model and the model of the cluster.

Hyperparameter tuning. Below the data visualization we display the current parameter values that influence the number of clusters and the detailedness of their explanations. Range sliders allow the user to change α, β, and the runtime limit of the greedy search. A more extensive look into the effects of these parameters follows in Section 5. Next to the parameter sliders we provide two ways of applying the new parameter values: *refine* and *recalc*. The refine option starts the algorithm from the current clustering as described in Section 3.2. The recalc option starts the algorithm from scratch. That way, users can either build on the current clustering and understanding of the data or investigate a new approach.

5 Experiments

In this section we discuss an empirical evaluation that we conducted to test ExClus. First we present the results from case studies on three data sets, secondly we consider quantitatvely the effects of the hyperparameters, and finally we discuss experiments on the scalability of the method.

5.1 Use cases

This section gives a short evaluation of the algorithm's results to give an initial insight into the application's possibilities.

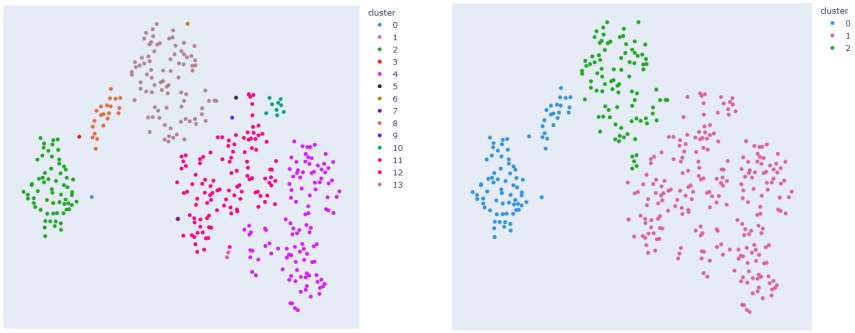
UCI Adult. The UCI Adult data set is an extraction from a 1994 USA census database, and the attributes included are: age (continuous), gender (male/female), ethnicity (white/other), education level (continuous), hours worked per week (continuous), and income ($\leq$50k, >50k). This experiment uses a sample of only 2500 of the nearly 32000 data points for performance reasons. An example of the algorithm's result is visible in Figure 1.

Dissecting the clustering reveals that the dimensionality reduction method mostly used three attributes to create the clusters: gender, income, and ethnicity. However, by tuning the hyperparameters, the algorithm can split these clusters further to reveal more details on the people included in each group or put clusters back together to show they are part of a more general pattern. For example, cluster two (green) contains all the white males with an income above \$50k, which is more than 30% of the data set. There is a possibility of splitting this cluster further to distinguish between attributes such as education level or hours worked per week by reducing β. However, this can overwhelm a user initially, so it is better to start with few clusters and then refine them to learn more details.

Other significant sections in this clustering are cluster zero (blue), which are the female counterparts of cluster two, and this cluster's explanation is also partially shown in Figure 1. The top right then contains all the high-income white people, with the red cluster being females and the other two (black and magenta) males, where the algorithm further separates them on high education level (black) and average education level (magenta). This difference also reveals that the dimensionality reduction does not reveal every pattern because there is no such split for the females as there are not enough high-income females included for the dimensionality reduction method to emphasize it. Finally, the bottom right has three clusters, including only other than white ethnicities, and the algorithm split them in almost the same manner as the white people, but again with less detail as there are not that many of them in the data sample.

German socio-economic data. This data set includes the socio-economic information of 412 German districts [2]. There are more than thirty attributes, but many of them are related, and overall, there are three main categories. First, the voting record attributes contain the voting percentages for the five largest parties (Green, Left, CDU, SPD, and FDP) in the 2005 and 2009 elections, where we included only the latter attribtes. Another block of attributes define the age distribution for the district, such as the percentage of old or young people. The remaining attributes give information on the workforce, such as which percentage of people work in which sector (agriculture, finance, service, etc.).

The application's outcome for two different parameter settings, shown in Figure 3, highlight a challenge in making ExClus effective. Notably, for hyperparameter settings that gave solutions with only a few clusters on other data, the solution here has two clusters with only a tiny number of data points (cluster zero and one). On the UCI Adult data this happened only with extreme parameter settings. Changing the hyperparameters here worsens this issue. On the one hand, a user could solve this by changing the hyperparameters of the dimensionality reduction method, t-SNE in this case, but it can be a strenuous task to test different embeddings on the application each time. Furthermore, assuming that the low-dimensional representation is given, we would prefer that the application deals with this problem regardless of the embedding. It appears the problem is at least partly related to the dendrogram of the agglomerative hierarchical cluster. Specifically, the problem disappears if we consider another linkage criterion, where we used single linkage in other cases, here complete or



(a) Single linkage (b) Complete linkage

Fig. 3: Results from the ExClus algorithm on the German socio-economics data for hyperparameter values α 250, β 1.6 and different linkage criteria.

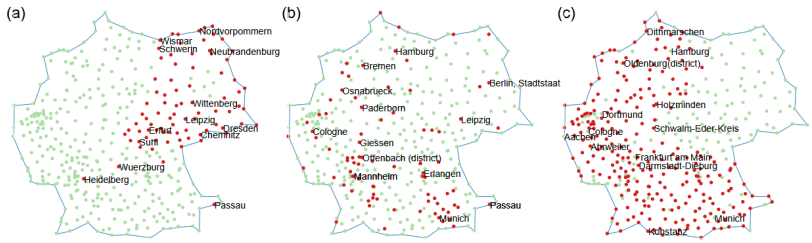


Fig. 4: Patterns discovered on the GSE data in the subjectively interesting subgroup discovery paper (Lijffijt et al. [9]). Reproduced with permission.

average appears to be preferable. The single linkage appears to fail here because the clusters are not as homogeneous as for other data sets.

Interestingly, looking at the clusters in Figure 3b, and highlighting the districts on a German map, they align almost entirely with the patterns discovered in previous research by Lijffijt et al. in their research on finding subjectively interesting subgroups in data sets with real-valued attributes [9]. Comparing some of the attributes included in the explanations of the clusters further proves that these patterns are almost equal. Take, for example, cluster two (green), which maps to pattern b in Figure 4. Just as in the paper, this cluster consists of districts with a higher population density, indicating they are cities, with a political favor towards the Green party and a large percentage of middle-aged people.

Cytometry data. Cytometry measures the characteristics of cells and has a broad range of applications. In this case, the experiment uses single-cell data of one mouse also used in previous research, which looked at different ways to make sense of increasingly higher dimensional data retrieved with a specific cytometry technique [13]. Each cell is described by nine markers and the researchers in

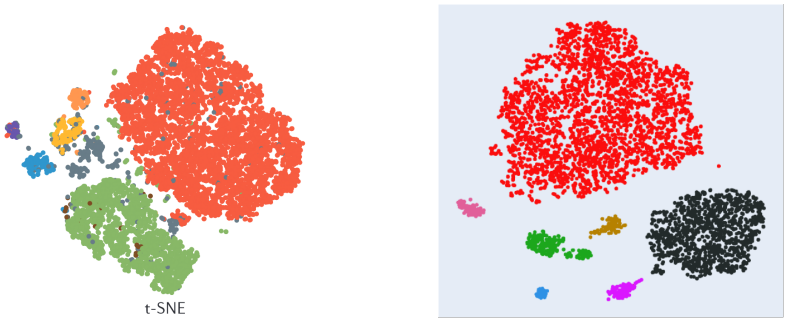


Fig. 5: (left) t-SNE dimensionality reduction on the cytometry data. Type of cell indicated by color. Obtained from the original paper [13]. (right) ExClus' clustering output on the Cytometry data set with hyperparameters α 250, β 1.2.

question used dimensionality reduction and clustering methods, which makes it ideal to evaluate ExClus and compare results.

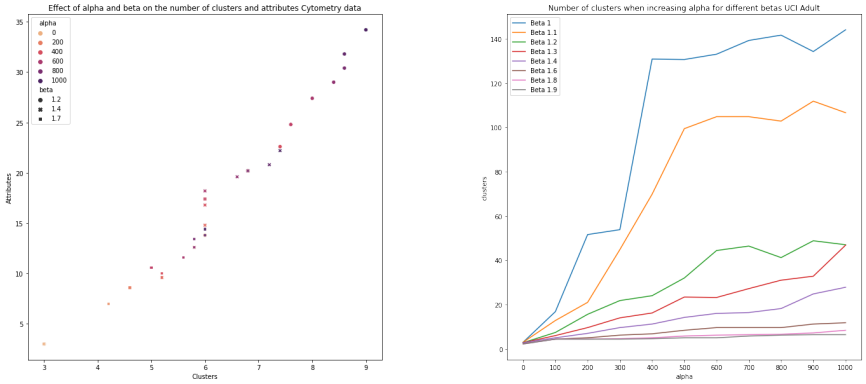
Figure 5(left) shows the results directly obtained from the original research paper. This dimensionality reduced version of the data set, calculated with t-SNE, has colored labels for each data point. The markers indicate which type of cell it is. These results are similar to the ones obtained from ExClus (Figure 5, right). The two large clusters are immediately distinguishable as similar and the smaller clusters also align almost perfectly. For example, the blue cluster from the ExClus results corresponds to the purple cluster from the original paper and the magenta cluster corresponds to the blue cluster.

Furthermore, ExClus simplifies the explanations. In the original cytometry paper color maps, showing the expressions of each gene marker, allows for analyzing the dimensionality reduced clustering. On the other hand, ExClus selects the most important attributes and represents them as a distributions compared to the data set's average. For example, to evaluate the blue cluster using the original method a user would have to examine all nine color maps while the ExClus algorithm only presents two attributes.

This case study shows that ExClus can explain the data set to the same extent as the paper, but with a lower descriptional complexity as a user does not need to scan all the different marker figures, which eventually simplifies and speeds up the process of understanding a data set.

5.2 Hyperparameters

Both parameters affect the number of clusters and detailedness of explanations, but they do this differently. α is a startup cost for the description length allowing for a more detailed explanation and more clusters as it results in the need for a larger description length before having a substantial impact on the ratio. This parameter's value has values between zero and a thousand. On the other hand, β serves as a penalty for the description length and usually has a value between



(a) Number of attributes and clusters for various α (color) and β (marker symbol) on the Cytometry data.

(b) Number of clusters for various α (x-axis) and β (colored lines) on the UCI Adult data.

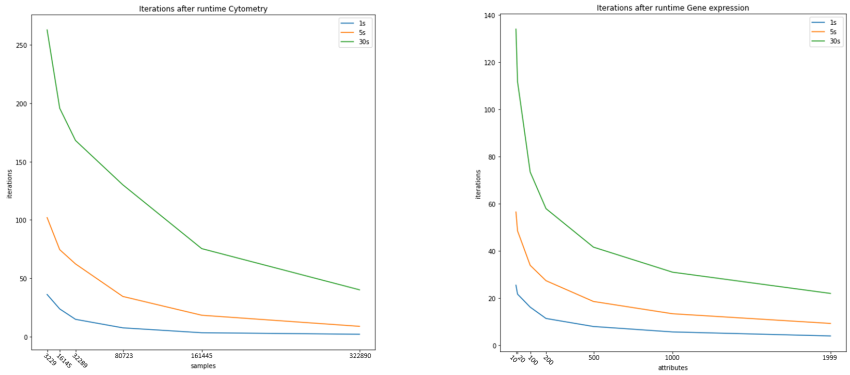
Fig. 6: Effects of varying the hyperparameters α and β.

one and two. If the value is larger than one, this applies a penalty that increases super-linearly with an increasing description length, forcing a faster cutoff on explanation and clustering.

Figure 6a shows the effect of α and β on the number of clusters and attributes. This graph reveals an almost linear increase in the number of clusters and attributes, where β divides the values into different sections, and α ensures a further evolution within each section. Furthermore, Figure 6b is a testament to why these parameters are necessary. It presents the number of clusters (y-axis) in the clustering the ExClus algorithm decided on for different values of α (x-axis) and β (color). If α is zero, there are almost no clusters, and changing β does not affect the results. Besides that, if β is one (no penalty), α changes the results almost uncontrollably (blue graph). While α and β might not have a straightforward relationship and interaction, they are both necessary and allow users to change the number of clusters and the number of attributes.

5.3 Scalability

Data sets can scale in two dimensions. On the one hand, the number of data points can increase, and on the other hand, the number of features can increase. Both can have severe effects on the runtime, and they should be taken into consideration whenever the algorithm runs. The algorithm goes through multiple iterations of the greedy optimization step. Each iteration searches for the optimal clustering with the number of clusters equal to the iteration. When the runtime ends, it selects the iteration's outcome with the highest subjective interestingness. Therefore, it is best to define scalability in the number of iterations the algorithm reaches. Figure 7a presents these effects when the number of data points increases and Figure 7b when the number of features increases.



(a) Number of iterations vs. number of *data points* on the Cytometry data.

(b) Number of iterations vs. number of *features* on the Gene expression data.

Fig. 7: Number of iterations the algorithm reaches within specified runtimes.

While both figures show an exponential decrease in iterations, this only becomes a noticeable issue if the data set is enormous (more than 100,000 data points or more than 500 features). Furthermore, in most cases, some sampling or feature importance selection will occur if the data set is this large because dimensionality reduction methods and hierarchical clustering techniques can also suffer from scalability issues. Besides that, it is unlikely for these data sets that a clustering with a vast number of clusters will be the optimal result as understanding more than a hundred patterns is highly complex. However, while it is not a limiting issue in most cases, the algorithm does have room for improvement in certain areas that could increase performance.

6 Conclusion

This paper introduced a new method to assist users in analyzing high-dimensional data sets. It contributes to the state of the art in considering how to find a good balance between informativeness and the complexity of information presented to the user, by making a trade-off between information and its descriptional complexity. Specifically, while generating clusterings and their accompanied optimal explanations automatically. The presented algorithm to identify an explainable clustering on top of a scatter plot of dimensionality-reduced data is implemented in a publicly-available tool called ExClus.

From case studies we have observed that ExClus can be used to effectively analyse data, by comparing results with previous studies on that data (German SE and Cytometry data), highlighting ExClus enables identification of previously known and possibly new patterns with little effort.

Further study could include investigation in methods to choose or support users in choosing good values for the hyperparameters. ExClus allows quick

experimentation with hyperparameters, but for now it remains an exercise of trial-and-error for users of the system. Secondly, we have omitted from the scope of this study which dimensionality-reduction method to use. We have used t-SNE in all experiments, which is not the most scalable algorithm and about which many critiques have been written. It would be worthwhile to investigate which dimensionality-reduction methods would synergize best with ExClus. Finally, we have observed that the linkage criterion can be important to consider. More generally, it may be interesting to study other ways to build a restricted search space like the agglomerative clustering approach used here.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Boley, M., Mampaey, M., Kang, B., Tokmakov, P., Wrobel, S.: One click mining: Interactive local pattern discovery through implicit preference and performance learning. In: Proc. of KDD-IDEA Workshop. pp. 27–35 (2013)
3. Cavallo, M., Demiralp, Ç.: A visual interaction framework for dimensionality reduction based data exploration. In: Proc. of CHI (2018)
4. Cavallo, M., Demiralp, Ç.: Clustrophile 2: Guided visual clustering analysis. IEEE TVCG **25**(1), 267–276 (2019)
5. De Bie, T.: Subjective interestingness in exploratory data mining. In: Proc. of IDA. pp. 19–31 (2013)
6. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
7. Faust, R., Glickenstein, D., Scheidegger, C.: DimReader: Axis lines that explain non-linear projections. IEEE TVCG **25**(1), 481–490 (2019)
8. Gunning, D., Aha, D.W.: DARPA's explainable artificial intelligence (XAI) program. AI Magazine **40**(2), 44–58 (2019)
9. Lijffijt, J., Kang, B., Duivesteijn, W., Puolamäki, K., Oikarinen, E., De Bie, T.: Subjectively interesting subgroup discovery on real-valued targets. In: Proc. of ICDE. pp. 1352–1355 (2018)
10. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. JMLR **9**, 2579–2605 (2008)
11. McInnes, L., Healy, J.: UMAP: Uniform manifold approximation and projection for dimension reduction. ArXiv:1802.03426 (2018)
12. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
13. Saeys, Y., Van Gassen, S., Lambrecht, B.N.: Computational flow cytometry: helping to make sense of high-dimensional immunology data. Nature Reviews Immunology **16**(7), 449–462 (2016)
14. Sedlmair, M., Brehmer, M., Ingram, S., Munzner, T.: Dimensionality reduction in the wild: Gaps and guidance. Tech. Rep. TR-2012-03, University of British Colombia, Vancouver, BC, Canada (2012)
15. Stahnke, J., Dörk, M., Müller, B., Thom, A.: Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. IEEE TVCG **22**(1), 629–638 (2016)
16. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)

COBRAS+: Reusing Previously Obtained Constraints in Active Semi-Supervised Clustering

Aras Yurtman[1 [0000−0001−6213−5427]], Wannes Meert[1 [0000−0001−9560−3872]], and Hendrik Blockeel[1 [0000−0003−0378−3699]]

KU Leuven, Dept. of Computer Science; Leuven.AI, B-3000 Leuven, Belgium

Abstract. Clustering is an inherently subjective process, where different clusterings of the same dataset may be desired in different applications. Semi-supervised clustering relies on partial ground truth information to obtain a clustering of interest. In the active setting, the clustering algorithm selects a query to maximize the return in each iteration. The longer the algorithm runs, the more constraints it obtains; however, even the best-performing algorithms do not consider the available constraints effectively. We propose an approach that exploits previously obtained constraints extensively in each iteration to avoid asking redundant queries and hence to decrease the number of queries that are needed to reach the same clustering quality. To be more specific, before asking a query to the user, we check for the existing constraints that are sufficiently similar to the selected query, and infer the response from them if possible. To assess the dissimilarity between constraints (and queries), we define a geometrical dissimilarity measure so that we reuse only relevant constraints and not the ones that may be misleading. We integrate our approach into the best-performing semi-supervised clustering algorithm, COBRAS, and name it COBRAS+. We demonstrate that, our approach decreases the number of required queries by more than 15% to achieve the same clustering quality on multiple publicly available datasets. We also show that our approach can more effectively use externally provided constraints in an incremental learning setting where the clustering algorithm starts with existing, externally provided constraints.

Keywords: Active learning · Semi-supervised clustering · Pairwise constraints.

1 Introduction

As an unsupervised learning problem, clustering is a popular technique in data analysis. It is inherently subjective, as for the same dataset there might be multiple clusterings of interest, depending on the application [10]. In the *semi-supervised* approach, there is limited user feedback, which might be in different forms such as labels of some instances or a set of constraints [19]. We consider *pairwise constraints*, each of which specifies whether two particular instances

should be in the same cluster or not; i.e., *must-link* (ML) or *cannot-link* (CL). Pairwise constraints suit the clustering problem better than labels of a subset of instances, because they are in line with what the user expects from the clustering problem: whether some instances are separated from each other or not. Moreover, in many applications it is easier to obtain pairwise constraints rather than class labels of instances. For example, instances in a query can be visually represented and the user only needs to answer as ML or CL; e.g., according to whether the animals in the given pair of images are of the same species or not.

User feedback is quite limited and costly in many applications; hence, should be leveraged to the greatest extent possible to maximize *query efficiency*. Equivalently, minimum number of queries should be asked to obtain a desired clustering quality. To this end, the clustering algorithm actively selects the pair of instances to be queried in *active semi-supervised clustering*. Furthermore, some algorithms have the *anytime* feature so that the user may stop the algorithm anytime and use the *intermediate clustering* that is available at that moment.

COBRAS is the most query-efficient constraint-based active semi-supervised clustering technique to date [17]. It forms a clustering and iteratively updates it by means of pairwise queries. It considers the transitivity and entailment properties to prevent redundancy in queries. However, it has a drawback that it does not extensively check previously obtained constraints before asking a query. This causes some of the queries to be similar to each other, which not only has an adverse effect on the query efficiency but also may annoy the user because he/she might think that his/her answers to previous queries are not considered by the algorithm.

The aim of this paper is to develop a technique that exploits existing constraints before asking a new query to the user in active semi-supervised clustering. A query can sometimes be *similar* to a previously asked one, the response of which we already know. In this case, we infer the response from the existing one instead of asking the new query to the user, which improves query efficiency at the risk of introducing noise. To compromise between wasting and overusing the available information, we define a measure to assess the dissimilarity between an existing constraint and the new query under investigation (Sec. 3.1), so that we infer the response only when the dissimilarity is below a predetermined threshold. We observe that the probability that randomly selected constraints are of the same type (ML or CL) increases when their dissimilarity decreases (Sec. 3.2). We evaluate our approach by integrating it into the COBRAS algorithm [17], and refer to the modified algorithm as *COBRAS+* (Sec. 3.3).

Our proposed technique can leverage both internally obtained constraints during the execution of the clustering algorithm and externally provided constraints; hence, we have two experimental setups. For the former, we compare the query efficiency and clustering quality of COBRAS and COBRAS+ (Sec. 4). For the latter, we start these two algorithms with a set of previously obtained constraints for a subset of the data (Sec. 5). We show that, in both cases, our proposed approach achieves the same clustering quality by asking fewer queries, or, a more accurate clustering by asking the same number of queries.

2 Related Work

The most straightforward way to develop a constraint-based active semi-supervised clustering algorithm is to extend an unsupervised clustering technique. The main approaches are to take pairwise constraints into account in the clustering process [19, 12, 20], in learning the similarity metric [21, 4], or both [2].

Most of the existing methods expect constraints to be provided to them in the beginning, and do not have the anytime feature [2, 3, 11, 18, 21]. Some of these methods query random pairs of instances [3, 21], decreasing the query efficiency. Active methods that select optimal queries, outperform them in terms of query efficiency [1, 2, 11, 18, 15–17, 22]. Some of these algorithms exhibit the anytime feature [22, 16, 17]. Among them, normalized point-based uncertainty (NPU) [22] is not time-efficient for large datasets, although it is query-efficient [17]. The remaining and more recent methods, COBRA [16] and COBRAS [17], are both query- and time-efficient while also having the anytime feature.

COBRA (constraint-based repeated aggregation) [16] relies on the concept of a *super-instance*: It first overclusters the dataset into the so-called super-instances by k-means clustering. Then, it obtains all of the pairwise relations between these super-instances by querying the user and assigns the super-instances to clusters. During this process, it exploits the transitivity and entailment properties of the constraints. In the resulting clustering, each cluster is comprised of one or more super-instances.

A more recent algorithm, COBRAS (constraint-based repeated aggregation and splitting) [17], has been shown to outperform other methods. Although it relies on super-instances as COBRA does, it is more query-efficient and requires less parameters. It starts with a single super-instance that contains the entire dataset. Then, it iteratively splits super-instances and merges them (i.e., assigns them to clusters). In the splitting step, it divides the largest super-instance into multiple super-instances according to the splitting level that is determined by queries asked within the super-instance. In the merging step, it asks further queries to assign the super-instances to clusters, similar to COBRA. As the number of queries increases, the dataset is modeled using a greater number of super-instances of smaller size, and the granularity of the clustering increases, in parallel with the number of queries asked. It is shown to be more query-efficient than the other algorithms on multiple datasets [17].

There is a significant drawback of these algorithms: They are too strict in checking for previously obtained constraints and do not make use of existing constraints that are similar to the relation that is under investigation. Hence, they may ask somehow redundant queries that are similar to those answered before. COBRAS suffers from this problem because the independent splitting and merging phases cause similar queries to be asked and it has no procedure to prevent this. We address this issue by developing a selective constraint checking procedure to minimize the redundancy in the queries, and use them more efficiently. We consider the query responses (that are provided by the user) as the ground truth, as done in the majority of machine learning problems except for some studies such as [13].

3 Extensive Use of Existing Constraints and COBRAS+

To exploit only relevant existing constraints, we first determine a measure to assess the dissimilarity between an existing constraint and a query under investigation (which can be considered as another constraint) (Sec. 3.1). We investigate whether two constraints that are similar to each other are more likely to be of the same type (ML or CL) or not (Sec. 3.2). Then, we describe our approach to reuse existing constraints based on their dissimilarity and integrate it into the COBRAS method to obtain the COBRAS+ algorithm (Sec. 3.3). Finally, we empirically analyze the reuse of existing constraints (Sec. 3.4).

3.1 Assessing the dissimilarity between constraints

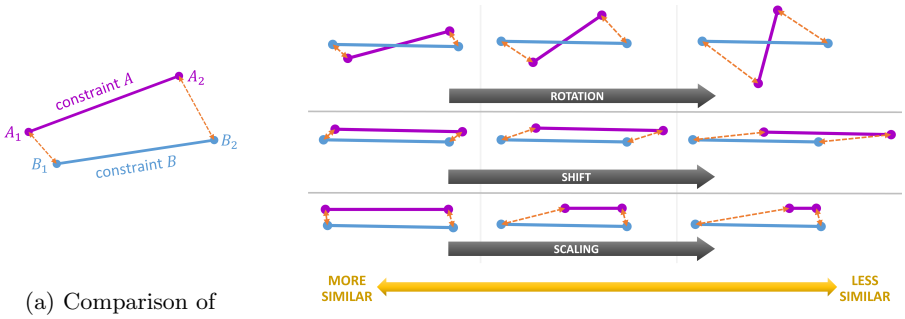
We need to calculate the dissimilarity between any two constraints, or between a constraint and the query under investigation. We develop our proposed dissimilarity measure geometrically and progressively as follows:

We consider each constraint as a line segment between two instances in the data space. An intuitive way to compare two constraints $A = (A_1, A_2)$ and $B = (B_1, B_2)$ is to calculate the Euclidean distance between their end points, $|A_1B_1|$ and $|A_2B_2|$, as shown in Fig. 1a. For the constraints to be considered as similar to each other, both endpoints should be close to each other, so we may calculate the dissimilarity measure as $\max(|A_1B_1|, |A_2B_2|)$. To take into account the fact that the constraints do not have any direction ($(A_1, A_2) \equiv (A_2, A_1)$), we may reverse one of them before the comparison and take the smaller of the two cases, having the dissimilarity of $\min[\max(|A_1B_1|, |A_2B_2|), \max(|A_1B_2|, |A_2B_1|)]$.

To apply a consistent threshold level to the dissimilarity measure across different datasets with different numerical ranges of features, we need to normalize the dissimilarity value. For this purpose, we use the average length of the two line segments that represent the constraints. Then, the dissimilarity measure between the constraints A and B becomes

$$d(A, B) = \frac{\min[\max(|A_1B_1|, |A_2B_2|), \max(|A_1B_2|, |A_2B_1|)]}{\frac{|A_1A_2|+|B_1B_2|}{2}} \quad (1)$$

Nine example pairs of constraints are shown in Fig. 1b. For each constraint pair, the two dashed lines between the end points show the distances that are used in the comparison, the shorter of them being the numerator of (1). The denominator is simply the average length of the blue and purple sticks. Rotation, shift, and shrinking of one (purple) constraint is illustrated while the other remains the same. In all of the three cases, we observe that the length of at least one dashed line increases from left to right in the grid; in addition, the length of one constraint decreases for scaling. Therefore, it is evident that the dissimilarity increases from left to right in the figure, as one might expect.



(a) Comparison of endpoints.

(b) Constraint pairs with different dissimilarity levels.

Fig. 1: Calculating the dissimilarity between constraints.

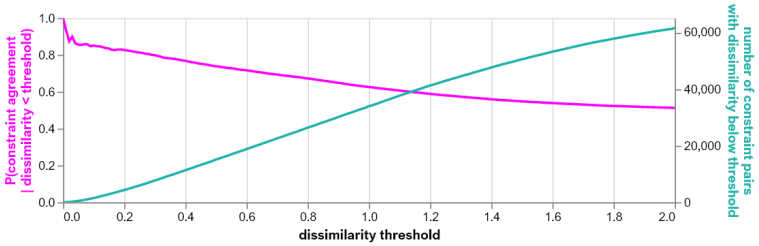


Fig. 2: Conditional probability that two random constraints are of the same type (ML or CL) given that their dissimilarity is below a threshold.

3.2 Evaluation of dissimilarity on randomly selected constraints

To analyze the relation between the dissimilarity between a pair of constraints and whether they are of the same type or not (ML or CL), we randomly select a number of constraints from real datasets (which are specified in Sec. 4.1). We make use of the true class information to randomly select constraints of a desired types. We have four cases for the types of constraint pairs; namely, ML-ML, ML-CL, CL-ML, and CL-CL. For each case, we select random instance pairs for the constraints with the restriction that the two constraints include an instance from a common class. This means that each ML-ML constraint pair is within the same class and the constraints of each CL-CL pair are associated with the same pair of classes. For every ML-CL pair, the ML constraint is within the class that is associated with one of the instances of the CL constraint.

The left vertical axis of Fig. 2 shows the conditional probability that two randomly selected constraints are of the same type given that the dissimilarity between them is below a threshold. This probability gradually decreases from 1 to around 0.5 when the dissimilarity increases from 0 to 2. Hence, there is a negative correlation between constraint agreement and the proposed *dis*similarity measure, as desired. The right vertical axis shows the number of randomly generated constraint pairs with a dissimilarity below a threshold.

3.3 Incorporating the constraint dissimilarity into the COBRAS algorithm: COBRAS+

To evaluate our approach, we build up on the COBRAS algorithm (which is originally presented in reference [17]), as it is the best-performing one in terms of query efficiency to date [17]. To obtain COBRAS+, we replace the querying function of COBRAS, which asks a query to the user each time it is called, by the proposed `analyze_relation` algorithm, which first checks the existing constraints and then asks the query if necessary. Below, we explain the COBRAS and COBRAS+ algorithms and then describe the `analyze_relation` function.

In *COBRAS* [17], instances are represented by super-instances, each of which is assigned to a cluster. Each super-instance has a representative, which is chosen as the medoid. As the algorithm runs, super-instances increase in number and decrease in size (i.e., they contain fewer instances); hence, the granularity of the clustering increases. Initially there is a single cluster that comprises a single super-instance which contains all the instances in the dataset. In each iteration, the largest super-instance is split into smaller super-instances. Then, the clustering assignments of super-instances are determined through queries formed between their medoids by considering the transitivity and entailment properties.

The COBRAS algorithm performs an exact constraint check before asking a query; that is, it checks whether a constraint between exactly the same pair of instances exists in the set of previously obtained constraints to avoid forming the same query. As super-instances are split into multiple smaller super-instances, their representative instances almost always change, and the previously obtained constraints no longer include the new representatives in general. Hence, COBRAS is very unlikely to find exactly the same constraint to rely on, so it almost neveruses previously obtained constraints.

In *COBRAS+*, we replace this exact constraint check by a more advanced process to reuse existing constraints according to a dissimilarity threshold τ that is specified by the user as a hyperparameter between 0 and 1. (Note that COBRAS+ is equivalent to COBRAS when $\tau = 0$.) This is implemented by replacing the `must-link` function in COBRAS by the `analyze_relation` algorithm, which is provided in Algorithm 1 and explained below.

The inputs of the `analyze_relation` algorithm are the pair of instances s_a, s_b, the existing constraints ($\mathbb{ML}$ and $\mathbb{CL}$), and the dissimilarity threshold τ. The pair (s_a, s_b) is associated with a pair of super-instances as well as a pair of clusters $(\mathcal{C}_a, \mathcal{C}_b)$, the relation of which is investigated. To determine this relation, a new query (s_a, s_b) is considered but not always asked, unlike what is done in the `must-link` function that is called by the original COBRAS algorithm [17]. Instead, in line 3 of Algorithm 1, we check for the previously obtained constraints (if any) between $\mathcal{C}_a$ and $\mathcal{C}_b$, and select the one (c^*) that is closest to $(\mathbf{s}_a, \mathbf{s}_b)$ in terms of the dissimilarity measure d defined in Sec. 3.1. If the dissimilarity is below or equal to τ (line 4), we infer the relation r between $\mathcal{C}_a$ and $\mathcal{C}_b$ from c^* (lines 6 and 8) and do *not* add $(\mathbf{s}_a, \mathbf{s}_b)$ to the set of constraints. Otherwise, all the existing constraints are above the threshold; hence, a new query $(\mathbf{s}_a, \mathbf{s}_b)$ is

asked to determine the relation r (line 10) and the query response is added in the set of constraints (lines 12 and 14).

3.4 Evaluation of previously obtained constraints in COBRAS

To observe to what extent we can reuse existing constraints, we analyze the the relation between existing constraints and the new query that is asked in an average COBRAS iteration in terms of the dissimilarity between them.

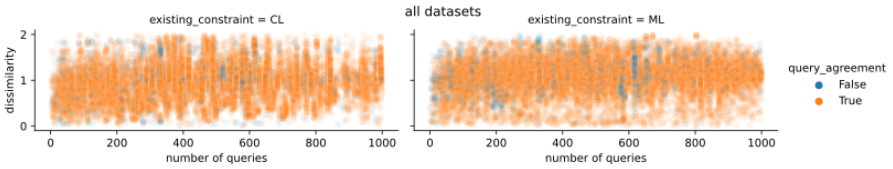
For each query COBRAS asks, we find the existing constraints between the clusters to which the instances in the query belong because the COBRAS+ algorithm checks the existing constraints that are between the same pair of clusters as the query to be asked. We show the scatter plot of these existing constraints according to their dissimilarity to the query to be asked and their type (ML or CL) in Fig. 3a. Although the scatter plots are obtained on multiple datasets (see Sec. 4.1), the dissimilarity value congregates around 1 and is on the same scale for all datasets (which are specified in Sec. 4.1) thanks to the normalization (the denominator of Eq. (1)).

Fig. 3b (left) shows the number of existing constraints below and above the dissimilarity threshold of 0.75 (see Sec. 6 for its selection) as a function of the number of queries. The number of existing constraints above the threshold increases as the algorithm runs. The right part of the figure shows that the ratio of using a previously obtained constraint (between the same pair of clusters) starts around 30% and gradually decreases to around 20%.

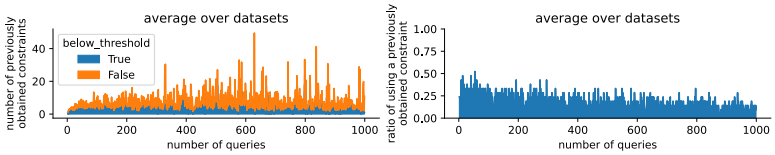
Algorithm 1: `analyze_relation`

Input: $\mathbf{s}_a, \mathbf{s}_b$: instance pair to query,
$\mathbb{ML}, \mathbb{CL}$: sets of previously obtained ML and CL constraints,
τ: threshold for constraint dissimilarity
Output: r: relation (ML or CL),
$\mathbb{ML}, \mathbb{CL}$: updated sets of ML and CL constraints

1 $\mathcal{C}_a, \mathcal{C}_b$ = the clusters to which $\mathbf{s}_a$ and $\mathbf{s}_b$ are assigned
2 **if** $\mathbb{ML} \cup \mathbb{CL} \neq \emptyset$ **then**
3 $\quad c^* = \arg\min_{(\mathbf{s}'_a, \mathbf{s}'_b)} \{d((s_a, s_b), (\mathbf{s}'_a, \mathbf{s}'_b)) \mid \mathbf{s}'_a \in \mathcal{C}_a,\ \mathbf{s}'_b \in \mathcal{C}_b,\ (\mathbf{s}'_a, \mathbf{s}'_b) \in \mathbb{ML} \cup \mathbb{CL}\}$
4 $\quad$ **if** $d((s_a, s_b), c^*) \leq \tau$ **then**
5 $\quad\quad$ **if** $c^* \in \mathbb{ML}$ **then**
6 $\quad\quad\quad r = \text{ML}$
7 $\quad\quad$ **else**
8 $\quad\quad\quad r = \text{CL}$
9 $\quad\quad$ **Return** $r, \mathbb{ML}, \mathbb{CL}$
10 $r = \texttt{query}(\mathbf{s}_a, \mathbf{s}_b)$
11 **if** $r ==$ ML **then**
12 $\quad \mathbb{ML} = \mathbb{ML} \cup \{(\mathbf{s}_a, \mathbf{s}_b)\}$
13 **else**
14 $\quad \mathbb{CL} = \mathbb{CL} \cup \{(\mathbf{s}_a, \mathbf{s}_b)\}$
15 **Return** $r, \mathbb{ML}, \mathbb{CL}$



(a) Scatter plot of the dissimilarity of the existing constraints for each iteration.



(b) The number of existing constraints and the ratio they are reused, both plotted as a function of the number of queries for $\tau = 0.75$.

Fig. 3: Statistics about existing constraints that are calculated cumulatively over all the datasets that are described in Sec. 4.1.

4 Experimental Evaluation

In this section, we explain the procedure we follow to evaluate the effectiveness of our approach. We first explain the datasets, our experimental methodology, and the evaluation metrics. Then we comparatively present the clustering quality and query efficiency for our approach, COBRAS+[1], and the state-of-the-art technique, COBRAS. Please refer to [17] for a comparison with other methods.

4.1 Datasets and experimental methodology

To evaluate COBRAS+, we use 21 publicly available datasets (UCI ML Repository[2], [3, 14, 1]): *breast-cancer-wisconsin, column-2C, dermatology, ecoli, faces-expression-imagenet, faces-eyes-imagenet, faces-identity-imagenet, faces-pose-imagenet, glass, hepatitis, ionosphere, iris, newsgroups-diff3, newsgroups-sim3, optdigits389-full, parkinsons, segmentation, sonar, spambase, wine*, and *yeast*.[3] These datasets were used in the original COBRAS algorithm [17] and also in earlier studies on constraint-based clustering [3, 22].

We consider the true class labels of the datasets as cluster labels by ignoring their ordering. We execute the clustering algorithms COBRAS+ and COBRAS on the full datasets but we allow the queries to include only training instances. To determine the training instances, we apply 10-fold cross validation and repeat the whole process 5 times. We limit the number of queries to 1000.

[1] `https://github.com/aras-y/cobras`

[2] `http://archive.ics.uci.edu/ml`.

[3] There are four different set of class labels for the *faces* dataset, and two different subsets for the *newsgroup* dataset; hence, 21 clustering tasks, each of which is considered as a dataset in this text. Refer to [17] for details.

4.2 Evaluation metrics

To evaluate the methods in terms of their clustering quality, we use *Average Rand Index (ARI)*, which measures the similarity between cluster labels obtained by an algorithm and true cluster (or class) labels [7, 6]. To calculate ARI, first the Rand index is calculated, which is the probability that two randomly chosen instances agree on whether they are in the same cluster or not in the two clustering assignments. Then, ARI is obtained by correcting this probability by using its expected and maximum values [8]. ARI is 1 when the clustering is the same as the ground truth, expected to be 0 when it is randomly obtained, and less than 0 when it is worse than random.

COBRAS+ (and COBRAS) generates a clustering in every iteration, which is returned if the algorithm is stopped at the end of that iteration; hence, an ARI is calculated for every iteration. We average the ARI over the repetitions and cross-validation iterations (and for datasets in some cases). We always retain the dependence of ARI on the number of queries in order to show the progressive course of the algorithm and to evaluate the query efficiency.

To make it manageable to compare the ARI values of different algorithms (COBRAS and COBRAS+, possibly with different threshold values) on multiple datasets, we summarize the number of wins of the algorithms against the others. To this end, we calculate *average aligned rank* of the ARI values for every iteration as follows [9]: For each dataset, we calculate the average ARI over the algorithms and subtract it from the individual ARI values. We then average the ranks of the items associated with each algorithm in a sorted list of all ARI values (for every algorithm for each dataset). The lower the average aligned rank is, the better the clustering is.

4.3 Evaluation of the proposed approach

Fig. 4 depicts clustering results for the proposed approach COBRAS+ and for the COBRAS algorithm as a reference [17]. The threshold is selected as $\tau = 0.75$ for COBRAS+ (see Sec. 6). Parts (a) and (b) of the figure show the average ARI and aligned rank curves, respectively. COBRAS+ clearly outperforms COBRAS after 25 queries in terms of clustering quality that is represented by the ARI. After asking 25 queries, it improves the query efficiency by at least 15% on the average; that is, it requires at least 15% less queries to obtain the same ARI.

The ARI curves are shown for individual datasets in Fig. 4c. For the majority of the datasets, we observe a trend similar to the overall results; however, there are some exceptions: For example, the dataset *dermatology* is trivial to cluster for all threshold values, and both COBRAS and COBRAS+ performs equally well in practice. In contrast, *faces-expression-imagenet* is quite difficult to cluster in this scheme, and the ARI is close to 0, which corresponds to random clustering, for both of the algorithms. The main reason is the dimensionality being much larger than the number of instances. The *yeast* dataset is also clustered inaccurately for both of the algorithms, which is consistent with moderate accuracies obtained using other clustering methods [5].

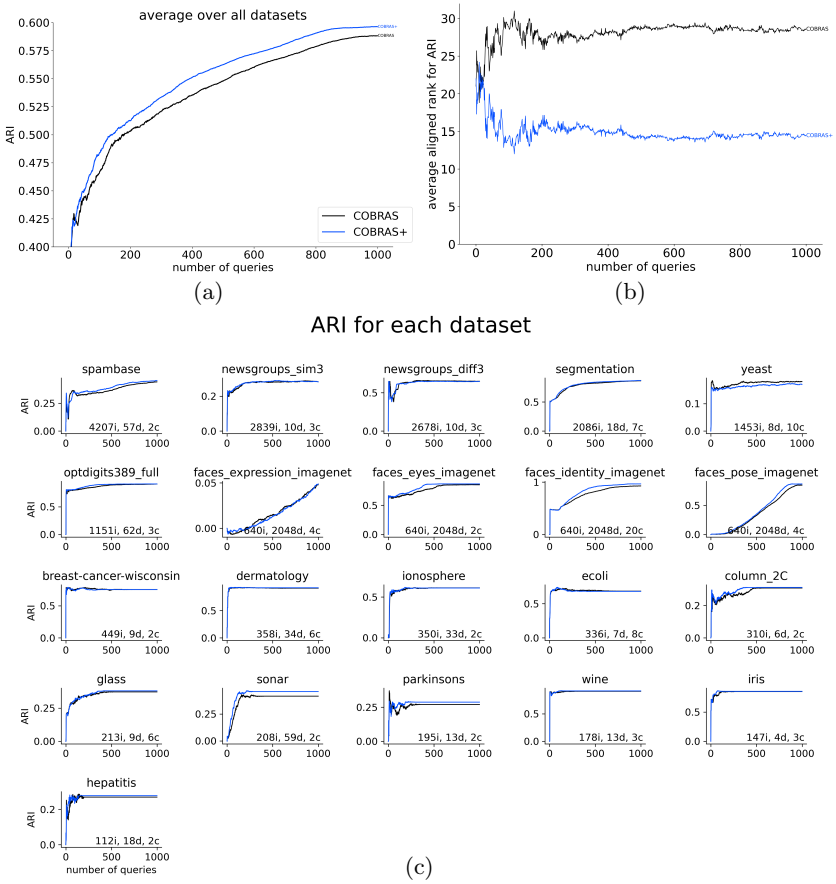


Fig. 4: (a) Average ARI and (b) average aligned rank as a function of the number of queries for COBRAS+ and COBRAS. (c) ARI for each dataset. The number of instances (i), dimensions (d), and classes (c) are provided at the bottom right corner of each plot.

5 Incremental Clustering: iCOBRAS+

In this section, we comparatively analyze the proposed algorithm COBRAS+ when there exists a set of pairwise constraints that were previously obtained based on (a subset of) the dataset and provided to the algorithm. This is an *incremental learning* problem because the algorithm starts with the previously obtained, limited information. When the COBRAS and COBRAS+ algorithms are run in this incremental setting, we call them iCOBRAS and iCOBRAS+, respectively. We expect iCOBRAS+ to make use of externally provided constraints to a greater extent and obtain a more accurate clustering than COBRAS by asking the same number of new queries.

We apply the following steps to evaluate incremental clustering for a given dataset D:

1. We split D into two parts: D_1 and D_2. We consider the following methods to perform this, and analyze each method separately:
   - **50% stratified split (SS)**: We randomly split each class into two, and include them in D_1 and D_2. This way, we retain the proportions of instances of each class so that D_1 and D_2 have a similar distribution.
   - **50% split according to classes (SAC)**: We include instances associated with a random half of the classes in D_1, and the rest in D_2.
   - **Leave-1-Out (L1O)**: We include data associated with a randomly selected class in D_2, and store the remaining part in D_1.

   In SAC and L1O, we omit 8 of the 21 datasets that contain only two classes.
2. We execute the COBRAS+ and COBRAS algorithms on D_1 by limiting the number of queries by 250, and save the pairwise constraints that are obtained from the queries.
3. We execute the algorithms on the full dataset $D = D_1 + D_2$ by providing the constraints that are obtained in Step 2. We limit the number of newly asked queries by 1000. The algorithms may utilize these externally provided constraints as well as new constraints that are obtained gradually based on newly asked queries. Table 1 shows main properties of the relevant clustering algorithms about their reliance to existing constraints.

As in Sec. 4.1, we repeat the incremental learning process 5 times for each technique (iCOBRAS+ and iCOBRAS), and present the results for both Steps 2 and 3 as a function of the number of queries in Fig. 5–7 for the three splitting methods (SS, SAC, and L1O). For Step 3 (the right parts of the figures), we also execute non-incremental methods, COBRAS+ and COBRAS, that do not use externally provided constraints that are obtained in Step 2. Part (a) of these figures provides the ARI whereas Part (b) provides aligned average ranks (see Sec. 4.2). For iCOBRAS+ and COBRAS+, we select the threshold of 0.75 for the dissimilarity, as the non-incremental version.

The left parts of Fig. 5–7 are provided for completeness, and show the ARI and the averaged aligned rank for the non-incremental algorithms executed on the D_1 part of the datasets (Step 2). As expected, COBRAS+ performs better than COBRAS. 250 queries obtained (for each dataset) in this COBRAS+/COBRAS execution are fed to the iCOBRAS+/iCOBRAS algorithms that are executed on the full dataset $D_1 + D_2$ for which the results are provided in the right parts of the figures.

Table 1: Types of clustering algorithms.

method	COBRAS	iCOBRAS	COBRAS+	iCOBRAS+
uses externally provided constraints	✗	✓✗	✗	✓
uses constraints obtained within the execution	✓✗	✓✗	✓	✓

✓: yes ✓✗: partially ✗: no

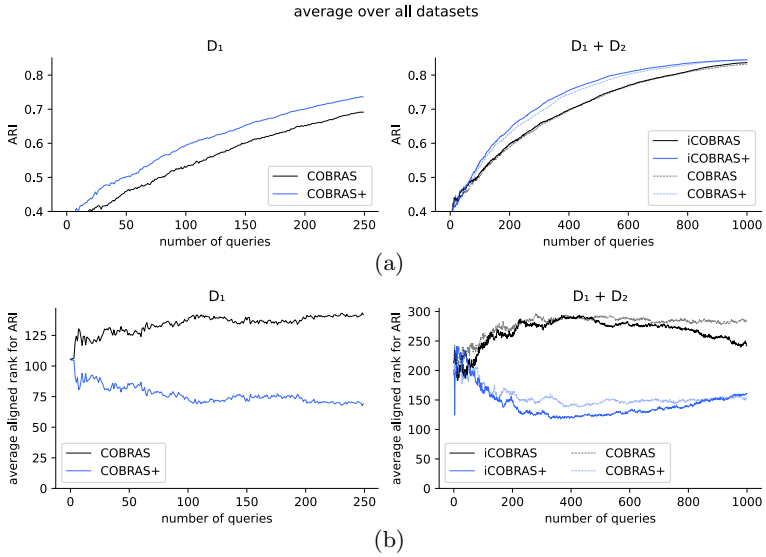


Fig. 5: (a) Average ARI and (b) average aligned rank plotted as a function of the number of queries for iCOBRAS with SS (stratified split).

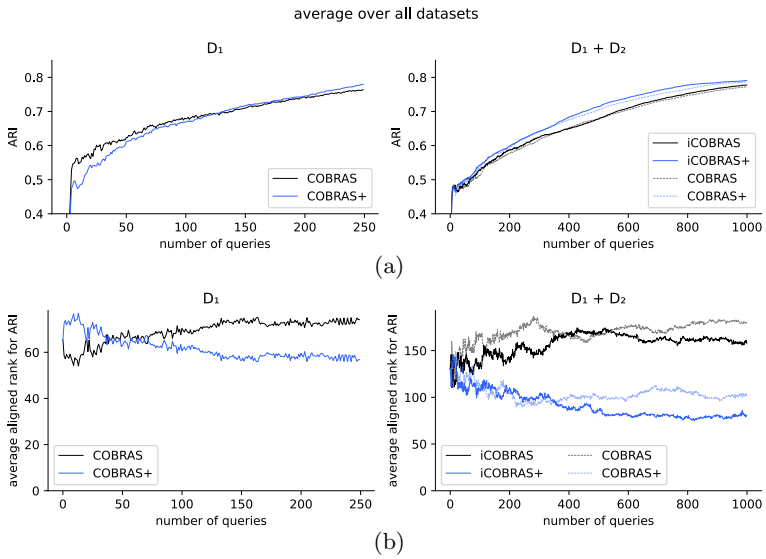


Fig. 6: (a) Average ARI and (b) average aligned rank plotted as a function of the number of queries for iCOBRAS with SAC (split according to classes).

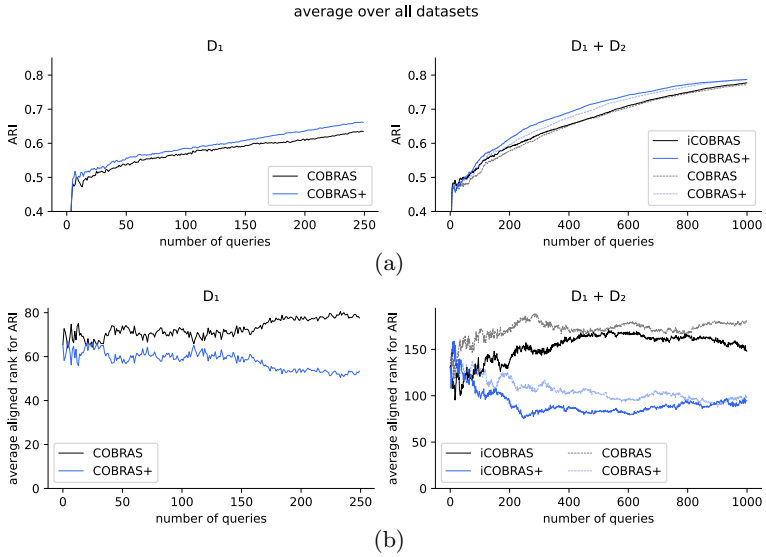
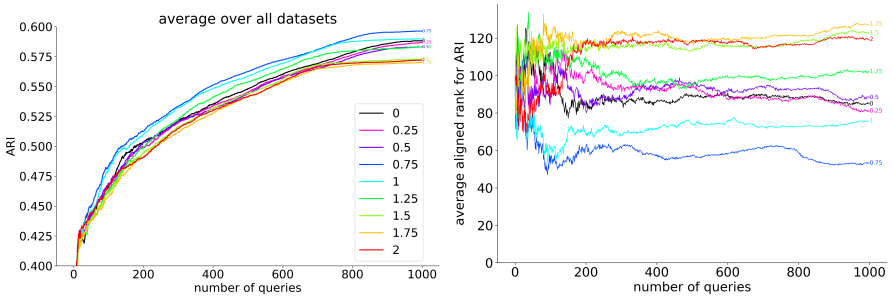


Fig. 7: (a) Average ARI and (b) average aligned rank plotted as a function of the number of queries for iCOBRAS with L1O (leave-one-out) split.

For SS (Fig. 5), iCOBRAS+ performs slightly better than COBRAS+ when the number of queries exceeds 120, mainly because from this point the clusters become pure enough (i.e., mostly contain instances from a single class) so that existing constraints between them often represent correct relations. In contrast, in the beginning, clusters are mostly mixed; hence, existing constraints between them are unreliable and iCOBRAS+ performs less accurately than iCOBRAS. For the splitting methods SAC and L1O (Fig. 6 and 7), the outperformance of iCOBRAS+ over iCOBRAS starts when the number of newly asked queries reaches 50, and is more consistent across the datasets, as the average aligned rank curves (in Part (b) of the figures) suggest.

For all of the three splitting methods, iCOBRAS and COBRAS perform similarly (see Fig. 5–7), because the (i)COBRAS algorithm cannot use previously obtained constraints unless there is an exact match, which is very unlikely to happen. The algorithms iCOBRAS+ and COBRAS+ perform much better than iCOBRAS and COBRAS because both of them can use existing constraints that are obtained during the execution of the algorithm, with the former being able to additionally use the externally provided constraints that are obtained in the previous execution on a smaller dataset (D_1).

Note that Fig. 5–7 provide results that are averaged over datasets. The same type of results are shown for individual datasets in Fig. 10–12 in the Appendix.



(a) (b)

Fig. 8: (a) Average ARI and (b) average aligned rank plotted as a function of the number of queries for COBRAS+. The numbers in the legend indicate dissimilarity threshold levels (τ). A threshold of 0 corresponds to COBRAS.

6 Sensitivity analysis for the dissimilarity threshold

We consider multiple threshold values for the constraint similarity for the COBRAS+ algorithm: $\tau \in \{0, 0.25, 0.5, \ldots, 2\}$ (see Sec. 3.3 for its use). For each threshold value, we plot the average ARI and the average aligned rank as a function of the number of queries in parts (a) and (b) of Fig. 8. ARI curves for individual datasets are provided in Fig. 9 in the Appendix. Note that higher ARI and lower rank are desired. The threshold of 0 corresponds to the original COBRAS algorithm [17].

Compared to COBRAS, selecting a small threshold does not change the ARI curve much. Noticeably higher ARI values can be reached with the threshold values of 0.75 and 1. As the threshold further increases, ARI decreases due to the overuse of previously obtained queries. Thus, COBRAS+ with an appropriately selected threshold (e.g. 0.75 or 1) considerably improves the clustering quality for the same number of queries. For this reason we have used the threshold level of $\tau = 0.75$ for the proposed (i)COBRAS+ algorithm throughout the paper.

7 Conclusion

We have proposed an approach to extensively leverage previously obtained queries in constraint-based active semi-supervised clustering. Considering that even the state-of-the-art approaches sometimes ask a query that is similar to a previous one, inferring the response from a similar existing constraint instead of consulting the user improves query efficiency and reduces the annoying behaviour of asking similar queries to the user. We have defined a dissimilarity measure between constraints and queries so that we can selectively exploit existing constraints based on their similarity to a new query to be asked. We have developed COBRAS+ by integrating our approach in the the state-of-the-art

technique, COBRAS, and improved its query efficiency. Compared to COBRAS, the proposed algorithm, COBRAS+, reaches the same clustering quality (ARI) by asking at least 15% fewer queries, or, equivalently, obtains a more accurate clustering by asking the same number of queries. We have also shown that the proposed algorithm provides higher ARIs in an incremental clustering setting where there are previously obtained queries obtained from a subset of the data.

8 Acknowledgements

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

1. Basu, S., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: Proceedings of the 2004 SIAM international Conference on Data Mining. pp. 333–344. SIAM (2004)
2. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 59–68 (2004)
3. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of the Twenty-First International Conference on Machine Learning. ICML '04, ACM, New York, NY, USA (2004)
4. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the 24th International Conference on Machine learning. pp. 209–216 (2007)
5. Fan, J., Feng, Z., Liu, W., Pang, J., Liang, Y.: Predicting yeast protein localization sites by a new clustering algorithm based on weighted feature ensemble. Journal of Computational and Theoretical Nanoscience **11**(6), 1563–1568 (2014)
6. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences **180**(10), 2044–2064 (2010), special Issue on Intelligent Distributed Information Systems
7. Hodges, J.L., Lehmann, E.L.: Rank methods for combination of independent experiments in analysis of variance. The Annals of Mathematical Statistics **33**(2), 482–497 (1962)
8. Hubert, L., Arabie, P.: Comparing partitions. Journal of classification **2**(1), 193–218 (1985)
9. Jr., J.L.H., Lehmann, E.L.: Rank Methods for Combination of Independent Experiments in Analysis of Variance. The Annals of Mathematical Statistics **33**(2), 482 – 497 (1962)
10. von Luxburg, U., Williamson, R.C., Guyon, I.: Clustering: Science or art? In: Proceedings of ICML Workshop on Unsupervised and Transfer Learning. Proceedings of ML Research, vol. 27, pp. 65–79. Bellevue, Washington, USA (02 Jul 2012)
11. Mallapragada, P.K., Jin, R., Jain, A.K.: Active query selection for semi-supervised clustering. In: 2008 19th International Conference on Pattern Recognition (2008)

12. Rangapuram, S.S., Hein, M.: Constrained 1-spectral clustering. In: Proceedings of the 15th Int. Conference on Artificial Intelligence and Statistics. Proceedings of ML Research, vol. 22, pp. 1143–1151. PMLR, La Palma, Canary Islands (21–23 Apr 2012)
13. Soenen, J., Dumančić, S., Blockeel, H., Van Craenendonck, T.: Tackling noise in active semi-supervised clustering. In: Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Springer (2020)
14. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
15. Van Craenendonck, T., Blockeel, H.: Constraint-based clustering selection. Machine Learning **106**(9), 1497–1521 (2017)
16. Van Craenendonck, T., Dumančić, S., Blockeel, H.: COBRA: A Fast and Simple Method for Active Clustering with Pairwise Constraints. In: Proceedings of the 26th Int. Joint Conference on Artificial Intelligence, IJCAI. pp. 2871–2877 (2017)
17. Van Craenendonck, T., Dumančić, S., Van Wolputte, E., Blockeel, H.: COBRAS: Interactive clustering with pairwise queries. Lecture Notes in Computer Science **11191**, 353–366 (2018)
18. Wagstaff, K., Cardie, C.: Clustering with instance-level constraints. AAAI **1097**, 577–584 (2000)
19. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: ICML. pp. 577–584 (2001)
20. Wang, X., Qian, B., Davidson, I.: On constrained spectral clustering and its applications. Data Mining and Knowledge Discovery **28**(1), 1–30 (2014)
21. Xing, E., Jordan, M., Russell, S.J., Ng, A.: Distance metric learning with application to clustering with side-information. In: Advances in Neural Information Processing Systems. vol. 15, pp. 521–528. MIT Press (2002)
22. Xiong, S., Azimi, J., Fern, X.Z.: Active learning of constraints for semi-supervised clustering. IEEE Transactions on Knowledge and Data Engineering **26**(1), 43–54 (2014)

Appendix A Supplementary Results

Fig. 9 shows the ARI curves for individual datasets for different dissimilarity threshold levels for COBRAS+. The average results over the datasets are given in Fig. 8.

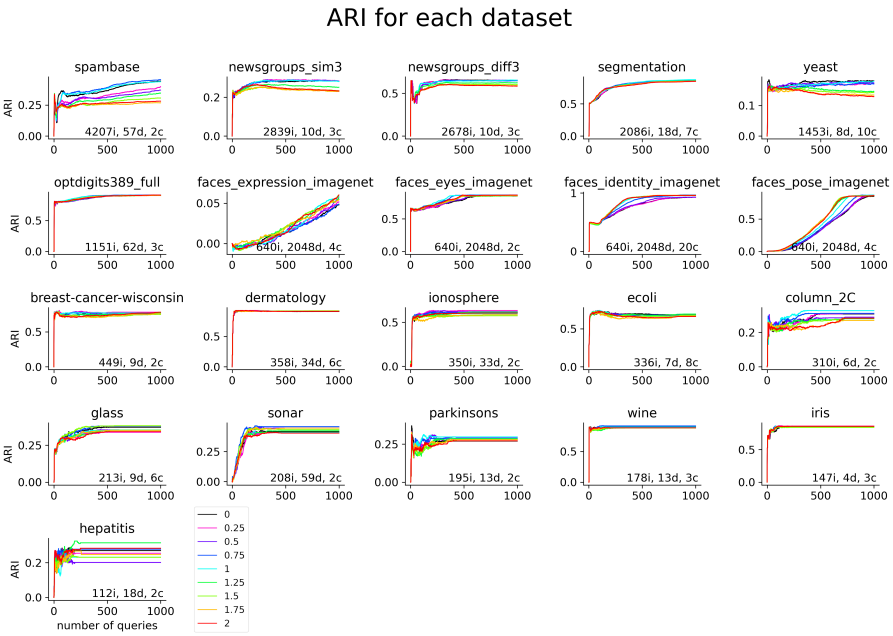


Fig. 9: Average ARI for different threshold levels for individual datasets. The numbers in the legend indicate the threshold levels. The number of instances (i), dimensions (d), and classes (c) are provided at the bottom right corner of each plot.

Fig. 10–12 show the ARI curves for individual datasets for iCOBRAS. Their averages over datasets are provided in Fig. 5–7, respectively.

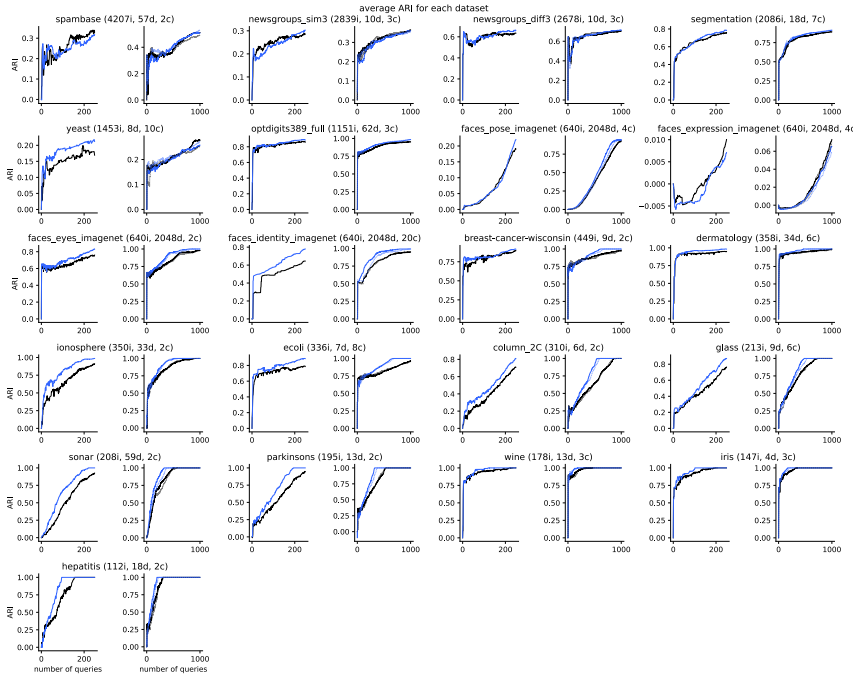


Fig. 10: ARI plotted as a function of the number of queries for each dataset for iCOBRAS with *stratified split*.

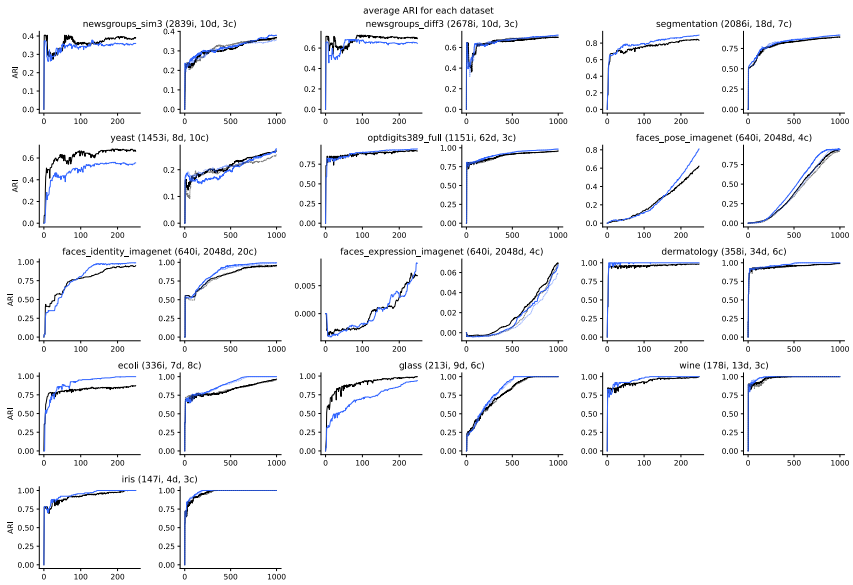


Fig. 11: ARI plotted as a function of the number of queries for each dataset for iCOBRAS with *class-based 50% split*.

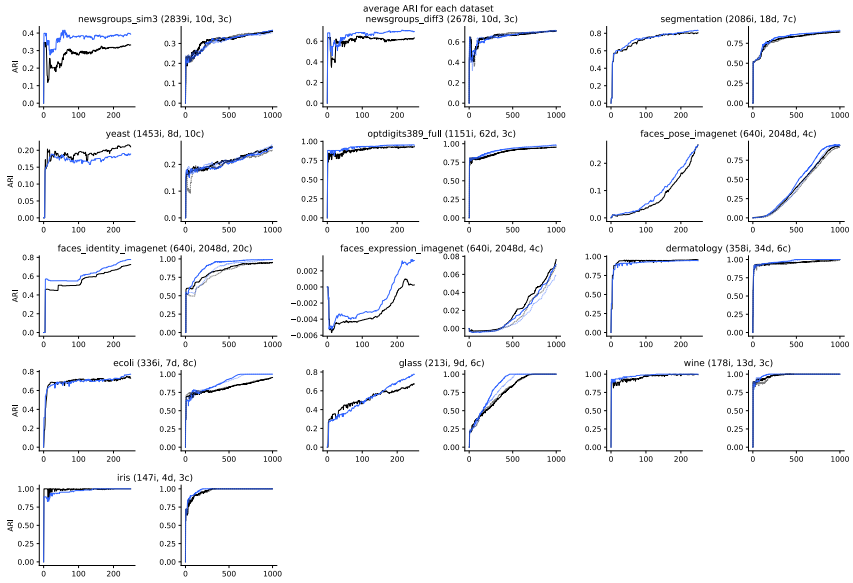


Fig. 12: ARI plotted as a function of the number of queries for each dataset for iCOBRAS with *leave-one-class-out split*.

Experiments of ASR-based mispronunciation detection for children and adult English learners*

Nina Hosseini-Kivanani[1,2][0000−0002−0821−9125], Roberto Gretter[3], Marco Matassoni[3], and Giuseppe Daniele Falavigna[3]

[1] University of Luxembourg, Luxembourg
[2] University of Trento, Trento, Italy
[3] University of Groningen, Groningen, Netherlands
[4] Fondazione Bruno Kessler (FBK), Trento, Italy
nina.hosseinikivanani@uni.lu, {gretter, matasso, falavi}@fbk.eu

Abstract. Pronunciation is one of the fundamentals of language learning, and it is considered a primary factor of spoken language when it comes to an understanding and being understood by others. The persistent presence of high error rates in speech recognition domains resulting from mispronunciations motivates us to find alternative techniques for handling mispronunciations. In this study, we develop a mispronunciation assessment system that checks the pronunciation of non-native English speakers, identifies the commonly mispronounced phonemes of Italian learners of English, and presents an evaluation of the non-native pronunciation observed in phonetically annotated speech corpora. In this work, to detect mispronunciations, we used a phone-based ASR implemented using Kaldi. We used two non-native English labeled corpora; (i) a corpus of Italian adults contains 5,867 utterances from 46 speakers, and (ii) a corpus of Italian children consists of 5,268 utterances from 78 children. Our results show that the selected error model can discriminate correct sounds from incorrect sounds in both native and non-native speech, and therefore can be used to detect pronunciation errors in non-native speech. The phone error rates show improvement in using the error language model. Furthermore, the ASR system shows better accuracy after applying the error model on our selected corpora.

Keywords: ASR · L2 learners · Detection of pronunciation errors · Computer−assisted pronunciation training (CAPT).

1 Introduction

The number of people who are learning a second language (L2) worldwide is increasing. Consequently, the need to evaluate and grade their pronunciation is becoming an important topic. Based on [10,16], in L2 learning progress, pronunciation plays an important role. However, this part of the learning process has

* Erasmus Mundus funded joint degree in Language and Communication Technologies (LCT).

always received less attention due to a lack of time and resources compared to other skills in classrooms [16]. The most challenging part of learning a new language is the pronunciation part because it is challenging to imitate sounds that are different from those of the native language's phoneme inventory [10,16,21]. [8] defines mispronunciation as surface pronunciation forms differing from canonical pronunciation forms. **Phoneme level mispronunciation** refers to the interference of a second language learner's native language during speech production, where foreign sounds are produced similar to a phoneme in their native language. Most of the classroom's pronunciation activities rely on the teacher to monitor, evaluate, and provide feedback on student pronunciation. This traditional technique does not seem adequate to correct student pronunciation, and it tends to be costly and time-consuming.

Given these constraints, the growing tendency to assess non-native languages leads to increased interest in automatic proficiency assessment of the speech. It boosts Computer Assisted Language Learning (CALL) tools in the field of language teaching for L2 learners [26,19,12]. CALL tools are designed to recognize words or sentences uttered by L2 learners by using an ASR system. The CAPT system is one of CALL's essential tools designed for automatically evaluating and detecting the learner's pronunciation errors. In the CAPT system, pronunciation evaluation can happen at two levels [20,21,25]: (i) detecting specific pronunciation errors, and (ii) an overall assessment of a speaker's proficiency (i.e., goodness of pronunciation (GoP) [29,5,28]). This study seeks to examine the use of ASR to empower learners to practice and improve their pronunciation on their own. Using a well-designed ASR system allows students to work autonomously while also offering flexibility to improve their pronunciation. The persistent presence of high error rates in speech recognition domains resulting from mispronunciations motivates us to find alternative techniques for handling mispronunciations. We use two corpora (see section 4) of both children and adult Italian speakers. These corpora contain both spontaneous and read-aloud tasks.

The innovation of our system is to use error rules in our language model (error language model) based on the most common errors seen in L2 learners' speech to improve our detection system (see section 3). Error detection of mispronounced input is similar to the process in which an annotator manually indicates pronunciation errors. Mainly, the topic of interest in this study is the mispronunciation assessment at the phone level. Ideally, our system should be able to detect errors the same as human annotators do.

2 Related Work

At the early stages of using ASR for pronunciation quality assessment, the preferred method was to consider the whole phrase without pointing to the error type and perform the assessment with the help of a hidden Markov model (HMM) [7]. In automatic pronunciation assessment, various features were evaluated such as speaking rate, articulation rate, phonation time ratio (for more information,

see [23]); apart from these features, ETS presents 29 candidate features [5] for scoring the speech of non-native learners of English.

Segmental features have been a subject of phonetics from the 1950s up to now [23]. The most common segmental features for investigating the pronunciation of L2 learners on automatic speech assessment are consonant features: stop closure duration, aspiration, and vowel features: vowel duration. In one of the first studies on the pronunciation assessment at a phone level, [27] found that the scoring accuracy for assessing errors was 80-92% for non-native English speakers (73 speakers).

Early research by [9] used two kinds of acoustic models to conduct automatic mispronunciation detection: (i) a model trained on native-speaker pronunciation, and (ii) a model trained on non-native speech. They used an acoustic model to calculate the log probability for each predicted phone in both models. They then calculated the difference between these two probability scores and used it as a metric for rating the pronunciation's quality. The result showed that the log-likelihood ratioLLR had a better performance than the log-posteriors method. According to [13], posterior probabilities and log-likelihood scores were the methods that were most correlated with word and phone level human assessments of pronunciation.

As an example of the earliest work in this field, we refer to the FLUENCY system [6] to detect pronunciation problems at both the phonetic and the prosodic levels (i.e., segmental and suprasegmental levels). They used the SPHINX-II speech recognizer to evaluate and detect phone errors and prosodic information (namely prosodic problems) of non-native speakers of French, German, Hebrew, Hindi, Italian, Mandarin, Portuguese, Russian, and Spanish learned English as their second language. It was reported that using ASR technology while learning a foreign language can reduce embarrassment and enhance learning for learners. [24] investigated three ASR systems for nativeness evaluation in their study: a GMM-HMM system, a DNN-HMM system, and a GMM-HMM system using DNN for feature extraction. The feature sets were categorized into fluency, rhythm, pronunciation, grammar, and vocabulary for segmental (at the consonant and the vowel level) and suprasegmental measurements of non-native speech in automatic non-native speech assessment. We will base our work on these features in our ASR system. The latest work related to the mispronunciation detection (e.g., [14]) tried to model uncertainty using a pronunciation model of L1 speech.

The above studies are a small set of examples that are related to segmental features. We focus on segmental features, in which we observe the specific phoneme-level errors made by second language learners. For learners, the most challenging part of learning a new language is the pronunciation part due to the fact that it is challenging to imitate sounds that are different to those of the native language's phoneme inventory [21]. Using a well-designed ASR system allows students to work autonomously while also offering flexibility that can lead to the improvement of their pronunciation.

[5] These features are related to the fluency or duration of silences of spoken English.

3 Methods

In the following sections, we describe the methods (language model and acoustic model) used to identify pronunciation errors made by Italian learners of English, to develop an ASR system to improve their L2 pronunciation.

3.1 Speech file transcriptions: Ideal, Manual, & ASR

For this study, the three transcriptions at the phone level were considered: Ideal, Manual, and ASR output. All the transcriptions are automatically time-aligned to the related speech signals:

- **Ideal (IDE)** (reference) refers to the canonical word's pronunciation.
- **Manual (MAN)** (reference) is the hand-corrected version of the ideal data at the phoneme level. Trained annotators corrected files manually using Praat tool [2], noting substitution, insertion, and deletion at the phoneme level. The manual output contains the manually corrected time-aligned phonetic transcription and can be considered the best possible transcription at the phone level.
- **ASR output** is the results of ASR processing given a set of phones as input (see section 3.3).

Given these three representations (i.e., IDE & MAN, IDE & ASR, and MAN & ASR), the comparison between specific pairs of phone sequences can provide us the following information:

IDE vs. MAN output: this gives us a true map of the errors made by Italian speakers when trying to speak English. Furthermore, the comparison of IDE & MAN could be used to build some models of the errors made by L2 speakers.; **IDE vs. ASR output**: this gives us a feeling of what an automatic system can detect for a new utterance, where no manual annotation is available; **MAN vs. ASR output**: this tells us how well an automatic system detects errors. Based on our purpose, the comparisons are as follows: MAN & ASR, IDE & ASR. The phoneme level matrix was computed as follows:

- Obtained the canonical (reference) phoneme level transcriptions of the speech data
- Obtained the phoneme level ASR output transcripts as a hypothesis (test) of the speech data, and the ASR output transcriptions were aligned with the canonical/reference transcriptions (here Manual and Ideal).
- Furthermore, the probability was computed per each phone.

The mispronunciation at the segmental level (i.e., phonetic) was categorized into three kinds at the phone level: substitution (i.e., a phoneme is replaced with another), insertion (i.e., an extra phoneme is inserted), and deletion (a certain phoneme is deleted) [21]. The following refers to the examples for each error type in the corpora. A sample of manual (MAN), ideal (IDE), and ASR annotations for the sentence "**I said white not bait**" are as follows:

Table 1: Sample transcription: Manually (MAN), Ideally (IDE), & ASR

MAN:	sil ay s eh d w ay t n oh t b ey t sil
IDE:	sil ay s ax d w ay t n oh t b ey – sil
ASR:	sil ay s ax d w ay t n oh t b ey – sil

Silences were marked as "sil".

The vowel /ax/ was substituted by the accurate vowel /eh/, and the consonant /t/ was deleted in ideal and ASR outputs. It means that the speaker mispronounced that vowel and replaced it with another vowel, close to the accurate vowel. The phone /t/ was not pronounced by the speaker, which refers to the deletion in this sample.

3.2 Sample of speech file

Generally, the sound wave representation of spoken words has two axes: time on the x-axis and amplitude on the y-axis. Figure 1 illustrates a sample sentence pronounced by a child, showing the speech waveform (top), the spectrogram (middle), and three text tiers (bottom) that report different segmental information in terms of words, phones, and some other information, along with their time boundaries. The spectrogram is a graphical representation which has three axes: time, frequency, and amplitude, where in 2-dimensional graphs, the amplitude is approximately visualized using a darker shading. Both corpora (ISLE and ChildEn) provided orthographic transcriptions at the word level for the speech input.

Figure 1 refers to the annotation and pronunciation of the phrase **"A birthday cake"** by an Italian male speaker. Three tiers were defined for each sound file; tier1) word, tier2) phone, and tier3) word score. Generally, three types of annotations were added to each audio for both corpora: sentence-level annotations (e.g., score and intonation), word-level annotations (e.g., pause), and phoneme-level annotations.

3.3 Proposed approach

Language model : The basic idea of language models is to provide a probability of a sentence or sequence of words; these probabilities are combined with the acoustic likelihood of the sequence and generate the resulting hypothesis [11]. In this work **phone n-grams** are used as a language model to allow the generation of phone sequences not constrained by only the words appearing in the lexicon; n-grams are trained on the phonetic transcription of the audio data with n ranging from 1 to 5.

Apart from the n-gram model, another novel approach is the application of an **error language model** based on the most frequent errors (hereafter error model). For the error model, we used lexical information by providing the

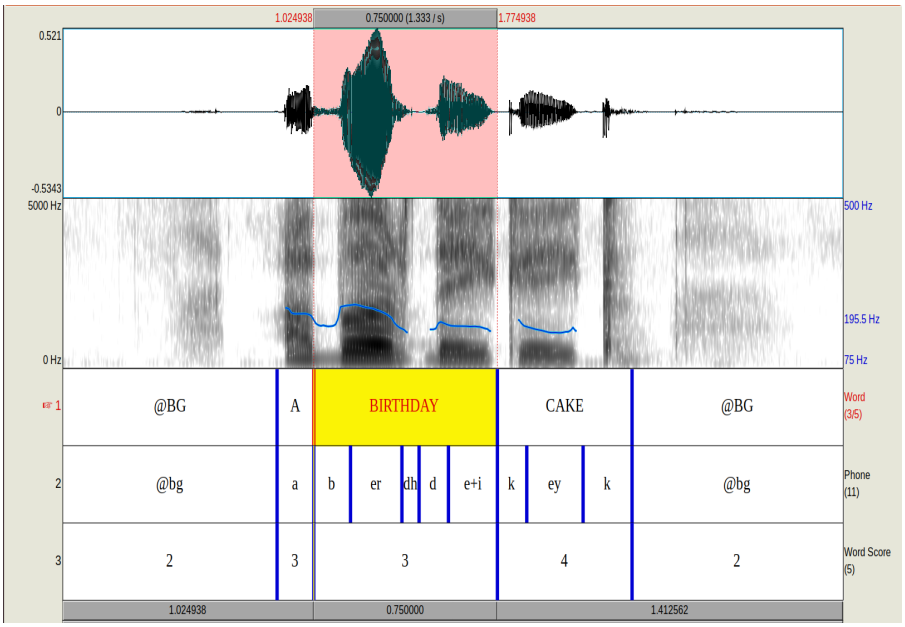


Fig. 1: A TextGrid of manual annotation example from ChildEn corpus. **Top:** Raw waveforms of (x-axis: time; y-axis: amplitude). **Middle:** Spectrograms (x-axis: time; y-axis: frequency; shading: amplitude (energy), darker means higher). **Bottom:** Word-level annotation of the signal.

ASR with the canonical pronunciations of the (known) words to be recognized and other pronunciations obtained by applying phonetic rules. These rules were defined manually by looking at the most common errors resulting from the 5-gram phone model (Table 3).

Acoustic Model: The acoustic model is learned from a set of audio recordings and their corresponding transcripts. We trained hybrid GMM-DNN models on English and Italian speech data from the child and adult Italian speakers. Based on the common Kaldi recipe [6], the selected acoustic models' features are: (i) initial GMM models trained on MFCC acoustic features with LDA transform and speaker adaptive training [18], (ii) use of i-vectors [15] of size 100, stacked to the MFCCs, and (iii) TDNNs trained using LF-MMI [22].

4 Data

The description of each corpus will be given below; (i) a corpus of Italian children (ChildEn) [7] [1], and (ii) corpus of Italian adults (ISLE) [17], who are learning

[6] Kaldi: `http://kaldi-asr.org/doc/`
[7] This corpus was designed and collected by ITC-irst: `http://www.itc.it`

English as their second language. Detecting mispronunciations in ASR requires corpora with labeling at the phonetic level: we selected these two corpora because they were manually labeled in terms of pronunciation quality by humans, and both were manually transcribed at phone level.

ChildEn consists of 5,268 utterances, with an overall duration of 3h:28m:26s from 78 children at about ten years of age. The selected students had been studying English at school for 3 or 4 years.

ISLE is a non-native speech dataset that contains utterances recorded by intermediate-level adult German and Italian learners of English. The audio files are 17h:54m:44s in total. The Italian section contains 23 Italian speakers.

4.1 Phone set

We created a phone list of each phoneme that includes canonical pronunciation and every possible mispronunciation of English phones by L2 learners; in other words, we have created a phoneme dictionary containing both English and Italian phones to be able to capture all possible mispronunciations. According to [3], Italian and English share only 40% of their phones. Therefore, we expect to see more phonological interference from Italian when L2 learners need to pronounce the phones in English. Moreover, in Italian, the relationship between spelling and pronunciation is straightforward compared to English. For example, in Italian, the letter 'a' is pronounced /a/, but in English, pronunciation and spelling are not strictly related to each other. For example the letter 'u' is pronounced /u/, /V/ or /3/ in English[8].

4.2 Evaluation

According to Figure 2, the most common mispronunciations for Italian speakers occur when the English target phone is not in the phoneme set of Italian (e.g., English phoneme /ax/). These phones are not contrastive for Italian speakers, leading to the speaker attempting to substitute the phonetically-closest phone from the Italian phoneme set or delete that phone. By "close" means that the acoustic signal has similar formants in both languages or the orthographic representation (spelling) is similar to spelling in Italian.

[26] provided a list of pronunciation errors that can be considered in ASR studies: Phonemic deletion, phonemic insertion, and phonemic substitution. For assessing the performance of a phoneme, the phone error rate (PER) will be used. The phone-error rate calculated the log-likelihood of a predicted phoneme given the acoustic signal [13,**?**]. The PER takes into account the errors related to phoneme substitutions (S), phoneme deletions (D), phoneme insertions (I), and P stands for the number of phones. If there are P phones in the reference transcript, and the ASR output has S, D, and I, then multiply by 100:

$$\text{PER} = \frac{\text{S} + \text{D} + \text{I}}{\text{P}} * 100 \tag{1}$$

[8] `http://archive.is/zsxA`

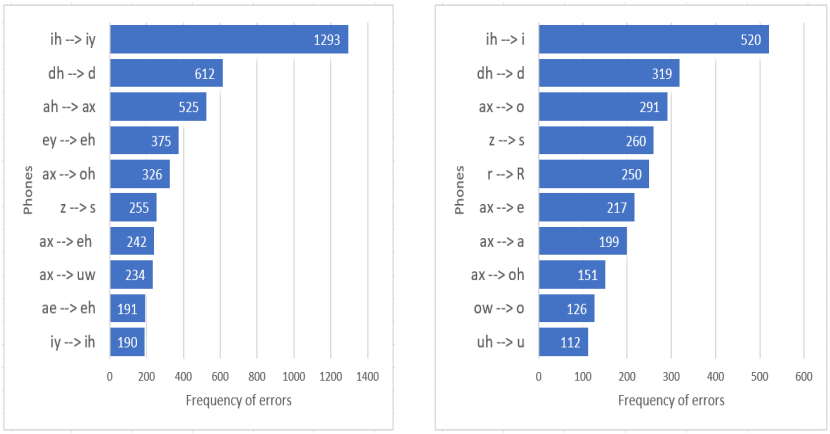


Fig. 2: Phoneme errors distributions of ISLE (left graph) & ChildEn (Right graph).

5 Results

5.1 Detection output

In this study, we are concerned with identifying mispronounced phonemes by L2 learners and improving our ASR system; we are interested in the PER and accuracy of our system.

We used Kaldi to perform speech recognition for phoneme error detection using the dictionary of words and phonemes. The acoustic model was trained with mixed speech data (i.e., children and adults), and the utterances were force-aligned based on the newly adapted word transcription. In our ASR system, we considered the following procedures in the acoustic model; (i) speech recognition based on the phone-level n-gram model, (ii) forced-alignment based on the existing word transcriptions, including the transcriptions modified using the mispronunciation data, and (iii) GMM classifier on using the native and the non–native acoustic model. The following sections compare the output of phoneme sequences to the gold standard, which leads us to identify pronunciation errors.

N-gram output The ASR was run on 1-gram, 2-gram, 3-gram, 4-gram, and 5-gram of phones. We first trained the algorithm introduced on our data to obtain the baseline values for the top frequent errors to determine the rules for the error model. According to [11], the common n-gram models, when there is sufficient training data, are trigram, 4-gram, or even 5-gram.

We trained all the n-gram models, but we saw that performance was much better with the 5-gram language model. For the rest of the n-grams, we only report the PERs for ASR & MAN, ASR & IDE, and IDE & MAN. The reason

is that the 5-gram model captures more context than other models, and thus we chose to report in full detail only the 5-gram results based on the number of substitutions, insertions, and deletions. For this reason, the 5-gram model was used for finding modified transcriptions. The PERs of 5-gram in ISLE and ChildEn are 42% and 38% for ASR & MAN comparison, respectively. As we have enough data for training, we considered the output of 5-gram models to develop our error model rules. The ASR systems' performance that used n-gram models was low; the PER for each n-gram model was between 33-52%.

Error model output Thus far, we have tried to improve the PER to estimate the ASR system's performance for mispronunciation recognition of L2 learners of English. The second model used the predefined rules (Table 3) from the n-gram model to train native and non-native speech data to check if the output of our ASR system will be improved compared to the n-gram model; we applied the forced-alignment by using the adapted lexicon. We compared our models' performance (n-gram model and error model) by calculating PER. We choose the best model based on the PER metric.

Interestingly, the new ASR system performs better in terms of PER for ChildEn, and the error rules showed better improvement for the ASR system. The speech recognition phone error rates are typically greater for adults than children. The low improvement in ISLE might be due to more complex and long sentences used in this corpus that lead to error propagation differently. The other possible reason might be due to the recordings' quality (i.e., noise in the background).

The purpose of this study is to find an ASR system with high accuracy for detecting mispronounced phonemes. Table **??** shows the statistics of error detection results comparing the error detected by our ASR system and errors reported by annotators for each corpus.

Table 2: Detection Accuracy: n-gram & error model

Corpora		**5-gram**	**error model**
ISLE	**ASR MAN**	58%	56%
	ASR IDE	62%	60%
	IDE MAN	88%	
ChildEn	**ASR MAN**	62%	72%
	ASR IDE	67%	76%
	IDE MAN	85%	

Overall, the highest Average Accuracy (see Eq. 1) to date was obtained using the error model. The accuracy results range from 72%-76% for correct error detection of phones in ChildEn (Figure 3).

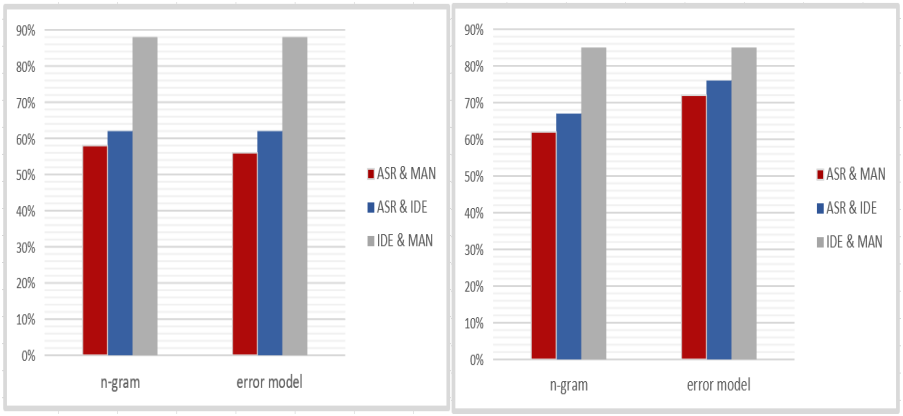


Fig. 3: Overall detection accuracy for ISLE (top graph). & ChildIt (bottom graph)

6 Discussion

Focusing on the smaller units would allow students to focus more on specific aspects of their pronunciation. However, the evaluation of smaller units for pronunciation assessment has a higher uncertainty compared to the evaluation of longer units [28]. For pronunciation training, especially for L2 learners, automatic pronunciation evaluation can play an important role, for example, by using ASR systems to evaluate the pronunciation of non-native input and quantify how close the speech is to a native-like pronunciation.

One of the challenges in using non-native speech for automatic recognition is the diversity of allophones, accents, invented words by L2 learners, and longer hesitations. Therefore, we identified high-priority phones (i.e., the phones used for creating the error rules (see Table 3)) and evaluated our system based on them. Since achieving a perfect native-like pronunciation is an unrealistic goal for adult learners, we focused on the specific mispronounced phones summarized in Table 3. As an example:

- **dh → dh/d**: "dh" can be replaced by itself or "d".
- **n d → n d/**: "d" can be deleted if it is in the sequence of "n d".
- **er → er r/R/**: means that our adapted lexicon1 will have three transcriptions for the word "HER" (i.e., "hh er" (ideal phonetic transcription), "hh er" R, & "hh er" r.

This model was trained on the phone errors observed in our corpora. We created a set of rules to consider these errors in our corpora and implemented our system based on the adapted lexicon. From the output of our ASR system, it is clear that the selected architecture is more capable of detecting mispronounced phones. In fact, our ASR systems' outputs showed that by applying n-gram and

Table 3: Error rules: Substitutions, Deletions, & Insertions

Substitution rules		**Deletion rules**		**Insertion rules**	
Original	Sub	Original	Del	Original	Insert
ih –>	ih/i/iy	n d –>	n d/	er –>	er r/R/
dh –>	dh/d	l d –>	l d/	aa –>	aa r/R/
ax –>	ax/a/o/oh	th d –>	th d/	ao –>	ao r/R/
z –>	z/s	s t –>	s t/	p l –>	p ax/o l
r –>	r/R	n t –>	n t/	b l –>	b ax/o l
uh –>	uh/u	ay k –>	ay k/	k l –>	k ax l
		ae k –	ae k	g l –>	g ax l

error models, the model proposed in this work obtained the lowest PER and better accuracy for both ISLE and ChildEn. In other words, in our system, the model that produced the best PER is the error language model with a PER of 23%, which is better than the rate in previous studies.

Previous works in speech recognition report that word error rate (WER) for human annotators on native data is around 5% compared to WER on non-native speech data, which is as high as 30% for automatic annotations [29]. Our ASR system may not perform better than this human-level performance from human annotators in detecting mispronounced phonemes in non-speech data. The best recent performance report on WER comes from an ETS project, in which the authors trained an ASR system on 800 hours of speech data, and the reported WER was 28.5% [4]. By considering the error language model, we can see that the ASR system performs better at recognizing the phones.

Furthermore, we can see that our ASR system selected the transcription defined in the new adapted lexicon based on our error model. Improving ASR will provide more speaking possibilities for learners to learn a new language. Using ASR can improve the traditional class to a more learner-centered environment with less anxiety. Apart from the segmental aspect, the suprasegmental aspect also plays a role in pronunciation training, but the ASR for the suprasegmental part is less reliable and still needs more work to improve it.

7 Conclusions

In this study, we worked on the language model - at the phone level - to extract the mispronounced phones and trained an acoustic model of native and non-native training examples to detect mispronunciations. We compared the PER of our ASR system using native and non-native speech data to determine our error model hypothesis's validity. Our system's innovation uses error rules in our language model (error model) based on the most common errors seen in L2 learners' speech to improve our detection system. Our error rules were defined based on the common errors of Italian speakers who learn English as their second

language. The error detection system records the learners' utterances, and the ASR-based detector will provide the phone-level transcription.

Finding errors is not the final destination. The next important step is to provide initiative feedback to learners which helps learners to correct their errors recognized by the ASR system. As the final step, a list of possible feedbacks can be given to L2 learners based on their pronunciation errors. Our ASR system needs to be adapted to the mother tongue of the L2 learners since different L1s can cause different pronunciation errors. Improving ASR will provide more speaking possibilities for learners to learn a new language.

References

1. Batliner, A., Blomberg, M., D'Arcy, S., Elenius, D., Giuliani, D., Gerosa, M., Hacker, C., Russell, M., Steidl, S., Wong, M.: The PF_STAR children's speech corpus. In: Ninth European Conference on Speech Communication and Technology (2005)
2. Boersma, P., Weenink, D.: Praat: Doing phonetics by computer (Version 6.0. 14). Retrieved from (last access: 29.04. 2018) (2016)
3. Browning, S.: Analysis of Italian children's English pronunciation. Report contributed to the EU FP5 PF STAR Project (2004)
4. Chen, L., Zechner, K., Yoon, S.Y., Evanini, K., Wang, X., Loukina, A., Tao, J., Davis, L., Lee, C.M., Ma, M., Mundkowsky, R., Lu, C., Leong, C.W., Gyawali, B.: Automated Scoring of Nonnative Speech Using the SpeechRaterSM v. 5.0 Engine. ETS Research Report Series **2018**(1), 1–31 (dec 2018). https://doi.org/10.1002/ets2.12198
5. Downey, R., Farhady, H., Present-Thomas, R., Suzuki, M., Van Moere, A.: Evaluation of the usefulness of the versant for english test: A response. Language Assessment Quarterly **5**(2), 160–167 (2008)
6. Eskenazi, M.: Using automatic speech processing for foreign language pronunciation tutoring: Some issues and a prototype. Language Learning and Technology **2**(2), 62–76 (1999)
7. Eskenazi, M.: An overview of spoken language technology for education. Speech Communication **51**(10), 832–844 (oct 2009). https://doi.org/10.1016/j.specom.2009.04.005
8. Fant, G.: Speech sounds and features. The MIT Press (1973)
9. Franco, Horacio and Neumeyer, Leonardo and Kim, Yoon and Ronen, O.: Automatic pronunciation scoring for language instruction organization. In: 1997 IEEE international conference on acoustics, speech, and signal processing. pp. 1471–1474. IEEE (1997)
10. Huensch, A.: The pronunciation teaching practices of university-level graduate teaching assistants of French and Spanish introductory language courses. Foreign Language Annals **52**(1), 13–31 (2019). https://doi.org/10.1111/flan.12372
11. Jurafsky, D., Martin, J.H.: N-gram language models. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (2018). https://doi.org/10.4324/9780203461891$_chapter_3$
12. Kawai, G., Hirose, K.: A call system using speech recognition to teach the pronunciation of japanese tokushuhaku. In: STiLL-Speech Technology in Language Learning (1998)

13. Kim, Yoon and Franco, Horacio and Neumeyer, L.: Automatic pronunciation scoring of specific phone segments for language instruction. In: Fifth European Conference on Speech Communication and Technology (1997)
14. Korzekwa, D., Lorenzo-Trueba, J., Zaporowski, S., Calamaro, S., Drugman, T., Kostek, B.: Mispronunciation detection in non-native (l2) english with uncertainty modeling. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7738–7742. IEEE (2021)
15. Madikeri, Srikanth and Dey, Subhadeep and Motlicek, Petr and Ferras, M.: Implementation of the standard i-vector system for the kaldi speech recognition toolkit. Tech. rep., Idiap (2016)
16. McCrocklin, S.M.: Pronunciation learner autonomy: The potential of Automatic Speech Recognition. System **57**(February), 25–42 (2016). https://doi.org/10.1016/j.system.2015.12.013
17. Menzel, W., Atwell, E., Bonaventura, P., Herron, D., Howarth, P., Morton, R., Souter, C.: The ISLE corpus of non-native spoken English. In: 2nd International Conference on Language Resources and Evaluation, LREC 2000. vol. 2, pp. 957–964. European Language Resources Association (2000)
18. Miao, Yajie and Zhang, Hao and Metze, F.: Speaker adaptive training of deep neural network acoustic models using i-vectors. IEEE/ACM Transactions on Audio, Speech, and Language Processing **23**(11), 1938–1949 (2015)
19. Neri, A., Cucchiarini, C., Strik, W.: Automatic speech recognition for second language learning: how and why it actually works. In: In Proc. ICPhS. pp. 1157–1160 (2003)
20. Neri, A., Mich, O., Gerosa, M., Giuliani, D.: The effectiveness of computer assisted pronunciation training for foreign language learning by children. Computer Assisted Language Learning **21**(5), 393–408 (2008)
21. Peabody, M.A.: Methods for pronunciation assessment in computer aided language learning. Ph.D. thesis, Massachusetts Institute of Technology (2011)
22. Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., Khudanpur, S.: Purely sequence-trained neural networks for asr based on lattice-free mmi. In: Interspeech. pp. 2751–2755 (2016)
23. Ryu, H., Hong, H., Kim, S., Chung, M.: Automatic pronunciation assessment of Korean spoken by L2 learners using best feature set selection. 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2016 (2017). https://doi.org/10.1109/APSIPA.2016.7820673
24. Tao, J., Ghaffarzadegan, S., Chen, L., Zechner, K.: Exploring deep learning architectures for automatically grading non-native spontaneous speech. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6140–6144. IEEE (2016)
25. Tejedor-García, C., Escudero-Mancebo, D., Cámara-Arenas, E., González-Ferreras, C., Cardeñoso-Payo, V.: Assessing pronunciation improvement in students of english using a controlled computer-assisted pronunciation tool. IEEE Transactions on Learning Technologies **13**(2), 269–282 (2020)
26. Witt, S.M.: Automatic error detection in pronunciation training: Where we are and where we need to go. Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training (IS ADEPT) **6**, 1–8 (2012)
27. Witt, S.M.: Use of speech recognition in computer-assisted language learning. Ph.D. thesis, University of Cambridge Cambridge, United Kingdom (1999). https://doi.org/10.1007/978-1-349-95810-8$_1$182

28. Witt, S., Young, S.: Phone-level pronunciation scoring and assessment for interactive language learning. Speech Communication **30**(2-3), 95–108 (feb 2000). https://doi.org/10.1016/S0167-6393(99)00044-8
29. Zechner, K.: What did they actually say? Agreement and disagreement among transcribers of non-native spontaneous speech responses in an English proficiency test. In: International Workshop on Speech and Language Technology in Education (2009)

Improving temporal smoothness of deterministic reinforcement learning policies with continuous actions

Bram De Cooman[1 [0000−0003−4843−3342]], Johan Suykens[1 [0000−0002−8846−6352]], and Andreas Ortseifen[2 [0000−0002−2555−4515]]

[1] KU Leuven, ESAT - STADIUS, Leuven, Belgium
{bram.decooman,johan.suykens}@esat.kuleuven.be
[2] Ford Research & Innovation Center, Aachen, Germany
aortseif@ford.com

Abstract. A commonly observed weakness of deterministic reinforcement learning policies with continuous action spaces, such as those obtained after training with the DDPG or TD3 methods, is the temporal roughness of their output signals (chosen actions). This is a serious deterrent for real-life application of such policies in continuous control tasks. For instance, in autonomous driving the rate of change of lateral acceleration is typically restricted to ensure passenger safety and comfort. Therefore, we propose a set of modified TD3 algorithms to improve the temporal smoothness of the trained agent's chosen actions. These smoothed TD3 (STD3) algorithms can be applied to smoothen policies; either in a post-processing training phase, or from the very start of training in an attempt to reduce the roughness cost (constraint) to an acceptable level throughout training. The proposed methodology is applied to some well-known benchmark environments, as well as to a more complex autonomous driving problem. Results show a consistent reduction of roughness without significant performance deterioration.

Keywords: Smooth Control · Reinforcement Learning · Deterministic Policies · Autonomous Driving.

1 Introduction

The usage of deterministic policies with continuous action spaces can lead to very oscillatory system behavior (see Figure 2). Such behavior is typically characterized by control signals with large, altering gradients in the time domain and high frequency components in the frequency domain. Although not as much of a problem in virtual simulations, this can severely impact the applicability of the learned policies in the real world, where such jerky control signals might wear down or damage critical components.

Mitigation. For simple or purely virtual environments such roughness issues could be dealt with using mitigation strategies. One option is to redefine the

action space and switch to derivative control of the system [4, 23, 17]. This way, the rough derivative actions will be smoothed out by the extra integrators in the environment's dynamics. Alternatively, the rough action signals could be low-pass filtered, effectively damping high frequency oscillations [8]. Another commonly used mitigation strategy in reinforcement learning consists of adding a roughness penalty in the reward signal [4, 23, 15], making it beneficial for the agent to select actions that do not change too much from one timestep to the next. While such techniques may work on relatively simple environments, they quickly become cumbersome in more realistic setups. In this paper, we try to tackle the roughness problem at its core, by embedding the smoothness constraints into the training process, leading to a set of smoothed TD3 (STD3) algorithms. These algorithms lead to smoother policies and simpler models (no unnecessary integrators and filters, less convoluted reward signals), allowing the designer to focus more on prime objectives instead.

Smooth exploration. Smoothness issues with neural network outputs have been addressed before [6] to increase the network's generalization capabilities. A recent overview of existing smoothing techniques for neural networks and their advantages is given by Rosca et al. [16]. In optimal control, the requirement of smooth control signals has been dealt with, e.g. under the form of slew rate constraints. It is thus surprising that such techniques have been rarely applied to the reinforcement learning domain. Initial attempts mostly focused on the smoothness of the exploratory policy during training. Lillicrap et al. [11] suggested the usage of autocorrelated Ornstein-Uhlenbeck noise to guarantee proper exploration of the state-action space when working with deterministic policies. Under small time discretizations, the rough uncorrelated Gaussian noise samples could cancel each other out, leading to insufficient exploration and suboptimal learned policies. Raffin et al. [14] presented generalized state-dependent exploration (gSDE) as another solution for the non-smoothness of Gaussian noise samples. By making the noise function state-dependent through a linear combination of policy features and fixing the linear weights for a given amount of training steps, the smoothness of the behavioural policy is drastically improved. The prime focus of such techniques lies on the smoothness of policies during training and exploration, while our focus in this paper lies on the smoothness of the learned policies during evaluation or deployment. Hence these methods could be seen as an orthogonal approach and could be readily combined with our proposed smoothed TD3 variants to improve the *overall smoothness*, during both training and evaluation.

Regularization. The usage of output regularization, in combination with derivative control, has been investigated by Chisari et al. [4]. By forcing the action rates to remain small, the integrated actions that are passed to the environment's dynamics remain smooth. Recently, Mysore et al. [12] introduced 'Conditioning for Action Policy Smoothness' (CAPS), a method to improve temporal and spatial smoothness of policies through the addition of two regularization terms on

the policy network. This smoothness regularization is also leveraged by our proposed STD3 methods, albeit in a more generic setting. In fact, the temporal smoothness regularization of CAPS corresponds to the specific $\text{STD3}_{\text{C,fix}}$ variant, introduced here. Spatial smoothness is not further considered in this work, but could also be accounted for using an extra regularization term and a dedicated spatial smoothing schedule.

This paper is organized as follows. In Section 2 the required reinforcement learning (RL) preliminaries are described, followed by a short motivational example in Section 3 to highlight the importance of additional smoothness constraints. Section 4 proceeds by introducing the different smoothed TD3 variants, used to improve the learned policy's temporal smoothness. Finally, the different variants are evaluated and compared on different environments in Section 5. These experiments show the great potential of the added smoothness constraints, as they not only drastically improve the policy's smoothness, but also outperform standard TD3 policies on a majority of the investigated environments.

2 Reinforcement Learning

In model-free Reinforcement Learning (RL) the objective is to find an optimal controller (policy) for an entity (agent) acting under a-priori unknown system dynamics (an unknown environment). At any given point in time t, the agent has access to the current state $\boldsymbol{s}_t \in \mathcal{S}$ of the environment; or an observation of this state if the system is only partially observable. The controller then maps these states $\boldsymbol{s}_t$ to suitable actions $\boldsymbol{a}_t \in \mathcal{A}$ and is often referred to as the agent's *policy*. The execution of an action, will bring the agent to a new state $\boldsymbol{s}_{t+1}$ — following the system dynamics — after which the same procedure can be repeated. To improve its policy, the agent has one extra source of information available: the reward signal $r(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ which describes how favourable it was to select action $\boldsymbol{a}_t$ while being in state $\boldsymbol{s}_t$ and transitioning to state $\boldsymbol{s}_{t+1}$. The optimal controller is thus the one which maximizes the agent's long term accumulated reward.

More formally, the RL problem can be described as the Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \sigma_0, \tau, r, \gamma)$ with state space $\mathcal{S} \subset \mathbb{R}^S$, action space $\mathcal{A} \subset \mathbb{R}^A$, initial-state distribution $\sigma_0(\boldsymbol{s}_0) : \mathcal{S} \to [0\,;1]$, state-transition distribution $\tau(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0\,;1]$, reward signal $r(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ and discount factor $\gamma \in [0\,;1)$. Note that the stochastic environment dynamics τ, σ_0 are modelled as a probability distribution but remain unknown to the agent. Usually, the agent's policy is also modelled through a probability distribution $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$ from which suitable actions can be sampled at every timestep. The special case of deterministic policies can also be considered $\boldsymbol{a}_t = \pi(\boldsymbol{s}_t)$. Learning the optimal policy π^* under such a framework then corresponds to finding the

policy maximizing the agent's *future discounted return* R_t at every timestep

$$R_t = \sum_{k=0}^{\infty} \gamma^k r(\boldsymbol{s}_{t+k}, \boldsymbol{a}_{t+k}, \boldsymbol{s}_{t+k+1}),$$
$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi,\tau,\sigma_0} \left[R_0\right]. \tag{1}$$

The notation $\mathbb{E}_{\pi,\tau,\sigma_0}$ is used to denote an expectancy taken over the probability distribution of actions $\boldsymbol{a}_t \sim \pi(\cdot|\boldsymbol{s}_t)$, induced by the policy, and over the probability distribution of states $\boldsymbol{s}_0 \sim \sigma_0(\cdot)$ and $\boldsymbol{s}_{t+1} \sim \tau(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t)$, induced by the environment. Although some RL methods try to directly search for the optimal policy using the objective (1), it is often useful to use (an estimate of) the policy's *action-value function* $Q_\pi(\boldsymbol{s}, \boldsymbol{a})$ for extra guidance

$$Q_\pi(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_{\pi,\tau} \left[R_t | \boldsymbol{s}_t = \boldsymbol{s}, \boldsymbol{a}_t = \boldsymbol{a}\right].$$

This action-value function satisfies following recursive relationship, known as the Bellman equation

$$Q_\pi(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_{\pi,\tau}[r(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) + \gamma Q_\pi(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}) | \boldsymbol{s}_t = \boldsymbol{s}, \boldsymbol{a}_t = \boldsymbol{a}].$$

A more extensive introduction to the domain of reinforcement learning is given by Sutton & Barto [21]. In this paper we will further limit the discussion to deterministic, off-policy, actor-critic methods, such as 'Deep Deterministic Policy Gradient' (DDPG) [11] and 'Twin Delayed DDPG' (TD3) [5]. These methods consist of two major components: the actor network $\mu(\boldsymbol{s}; \boldsymbol{\theta}_\mu)$ modelling the deterministic policy (state-action mapping) and the critic network $Q(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta}_Q)$ estimating the optimal state-action value function. Both components are jointly updated, improving one another as training progresses, using experience collected while the agent is interacting with the environment during training. As a deterministic policy maps the same state always to the same action, an external source of stochasticity is often required in order to sufficiently explore the state-action space. Hence, the behavioural policy $\beta(\boldsymbol{s}) = \mu(\boldsymbol{s}; \boldsymbol{\theta}_\mu) + \epsilon$ with exploration noise $\epsilon \sim N(\boldsymbol{0}, \boldsymbol{\sigma}_{expl})$ is used to collect experience during training instead of the deterministic policy modelled through the actor, making these methods *off-policy*. The collected experience tuples $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ are first stored in a replay buffer $\mathcal{B}$. In a second step, uniformly sampled batches of experience tuples from this buffer are used to update the actor and critic networks.

The critic network is updated by minimizing a squared temporal difference error[3]

$$L_Q(\boldsymbol{\theta}_Q) = \mathbb{E}_{(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1}) \sim \mathcal{B}} \left[\left(Q(\boldsymbol{s}_t, \boldsymbol{a}_t; \boldsymbol{\theta}_Q) - y_t\right)^2 \right],$$
$$y_t = r_t + \gamma Q(\boldsymbol{s}_{t+1}, \mu(\boldsymbol{s}_{t+1}; \boldsymbol{\theta}'_\mu); \boldsymbol{\theta}'_Q),$$

[3] For TD3 extra twin networks are introduced to avoid overestimation bias and an extra noise term is added to the target policy's actions to smoothen the value estimate [5].

where the primes on the weight vectors denote the usage of target networks to improve the stability of the learning process. The actor network is updated by minimizing the actor loss

$$L_\mu(\boldsymbol{\theta}_\mu) = -\mathbb{E}_{\boldsymbol{s}_t \sim \mathcal{B}} \left[Q(\boldsymbol{s}_t, \mu(\boldsymbol{s}_t; \boldsymbol{\theta}_\mu); \boldsymbol{\theta}_Q) \right], \tag{2}$$

leading to an approximation of the deterministic policy gradient [19]

$$\nabla_{\boldsymbol{\theta}_\mu} J \approx \mathbb{E}_{\boldsymbol{s}_t \sim \mathcal{B}} \Big[\nabla_{\boldsymbol{\theta}_\mu} \mu(\boldsymbol{s}; \boldsymbol{\theta}_\mu)|_{\boldsymbol{s}=\boldsymbol{s}_t} \nabla_{\boldsymbol{a}} Q(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta}_Q)|_{\boldsymbol{s}=\boldsymbol{s}_t, \boldsymbol{a}=\mu(\boldsymbol{s}_t; \boldsymbol{\theta}_\mu)} \Big].$$

3 Motivation

As a first, motivational example, the task of learning a simple overtaking manoeuvre on a three-lane highway is considered (Figure 1). At the start of each episode, the virtual driver (agent) is positioned in the rightmost lane behind a slower lead vehicle, travelling at a constant velocity. The virtual driver can perceive its current velocity components, the relative lateral offset w.r.t. its current lane center and the lanes directly to the left and right, and the relative offset and velocity components w.r.t. the car to overtake. The action space is two-dimensional, consisting of a longitudinal reference velocity and lateral reference offset (which are tracked by lower-level controllers). The reward $r(\boldsymbol{s}_t)$ is a weighted sum of two components $0.75r_V + 0.25r_R$. Where r_V is a penalty given when traveling at low velocities — thus rewarding policies which overtake the slow vehicle instead of staying behind it — and r_R is a penalty given when not driving in the rightmost lane (to obtain policies following common rules of the road). More details on the simulation environment and definitions of states and reward signals can be found in Appendix A.

Five policies (each initialized with a different seed) are learned in this environment using the TD3 method [5]. While each of them is able to correctly overtake the slow vehicle, therefore maximizing their long-term reward, only two of them do so in a smooth way. The others suffer from high-frequency oscillations in their lateral reference signals, severely impacting passenger comfort as

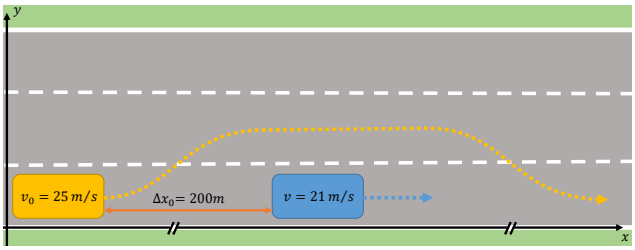


Fig. 1. Schematic overview of the motivational overtaking environment. The virtual driver (agent) is in control of the yellow vehicle and has to overtake the slower moving blue vehicle in front of it.

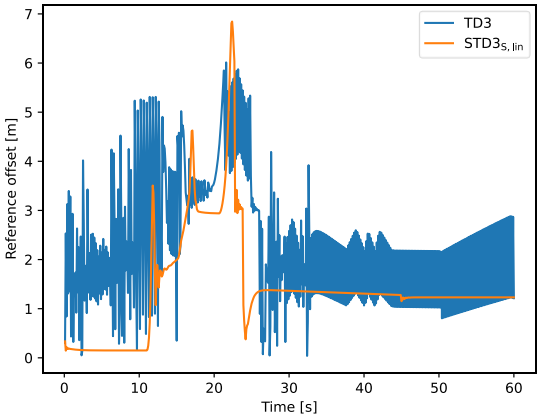


Fig. 2. Lateral reference offset of two policies trained for 250 episodes (with 600 timesteps) on the motivational overtaking environment. The blue line corresponds to the policy trained using the vanilla TD3 method, the orange line corresponds to the policy trained using the smoothed $\text{STD3}_{\text{S,lin}}$ variant (after 50 episodes of smoothing). While both policies correctly learn the overtaking manoeuvre (around $15 - 25s$), the rough reference changes of the TD3 policy prevent its usage in real vehicles.

illustrated in Figure 2. The occurrence of these oscillations throughout training is also quite volatile, as they seem to vanish and reappear within a few training episodes.

For simple environments, such smoothness problems could be dealt with by incorporating extra penalties in the reward signal and/or the usage of derivative control (see Section 1). However, this becomes increasingly more difficult for problems with more complex reward functions or state representations. Without proper care, the resulting policies could take a significant performance hit, as compared to their unconstrained counterparts (see discussion in Section 5.3 and Figure 6). The proposed smoothed TD3 variants in this work are more easily applicable and have higher robustness to such problems.

4 Methodology

To improve the smoothness of the learned policies, different smoothed TD3 (STD3) variants are introduced in the following subsections. First, a brief overview of the used roughness metrics is given.

4.1 Roughness metrics

Different metrics of smoothness or roughness of a curve or control signal exist:

– The integral of the second-order time-derivatives (or its approximation using sums and finite differences for discrete signals), as commonly used in smoothing spline applications [7].

– A metric defined over the frequency spectrum of the time signal, obtained after a Fourier transform, typically used in signal processing [6, 12].
– The average squared temporal difference of consecutive samples [14].

In the context of this paper, we use the third metric and calculate the average roughness of a discrete time signal $\boldsymbol{x}_k, 0 \leq k \leq k_M$ as

$$\bar{\rho} = \frac{1}{k_M} \sum_{k=1}^{k_M} \rho(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k). \tag{3}$$

The *immediate roughness* of the signal is then defined as

$$\rho(\boldsymbol{x}_0, \boldsymbol{x}_1) = \|\boldsymbol{x}_0 - \boldsymbol{x}_1\|_P^2 = (\boldsymbol{x}_0 - \boldsymbol{x}_1)^\top P(\boldsymbol{x}_0 - \boldsymbol{x}_1), \tag{4}$$

where P is a positive definite matrix that can be used to put more or less weight on certain signal components.

Equation (3) can be further generalized to time signals originating from sampling actions from a policy π under an MDP with finite episode length k_M. For the specific case of calculating the average roughness of the sampled actions we have

$$\bar{\rho}_\pi = \mathbb{E}_{\pi,\tau,\sigma_0} \left[\frac{1}{k_M} \sum_{t=1}^{k_M} \rho(\boldsymbol{a}_{t-1}, \boldsymbol{a}_t) \right]. \tag{5}$$

Different ways to approximate this expectancy will be given in the next subsection. Beside being a simple metric to calculate, this definition of roughness will turn out to be advantageous when combined with model-free reinforcement learning schemes.

Notice that taking $P = I$ gives the *unscaled roughness*, using the Euclidean norm of the action difference. Another commonly used choice for P throughout this paper is the diagonal matrix with elements $p_{i,i} = \Delta a_i^{-2}$ where Δa_i is the maximum absolute difference between the i-th component of two actions. This gives rise to the *immediate normalized roughness* ρ_{norm}, which is less impacted by action components with a larger range of possible values (i.e. with a higher Δa_i).

4.2 Smoothed TD3

We propose to modify the actor loss (2) by adding an extra weighted smoothness term, following existing smoothness regularization techniques for neural networks [6]. In this case, the smoothness term approximates the average roughness of the policy

$$R_{\text{smooth}} \approx \bar{\rho}_\pi.$$

The actor weights are then updated by minimizing the total loss

$$L_{\mu,\text{smooth}} = L_\mu + \lambda_s R_{\text{smooth}}. \tag{6}$$

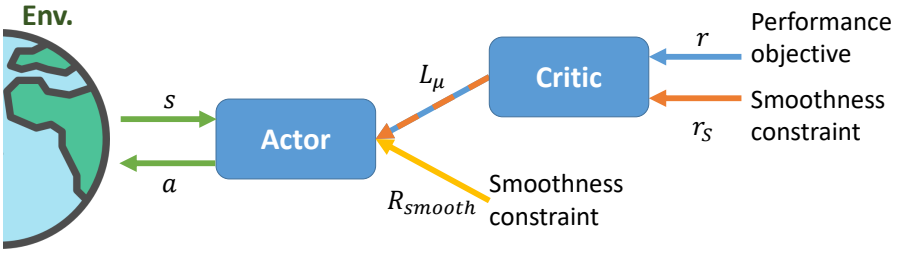


Fig. 3. Schematic overview comparing the information flow of the applied smoothness constraints using reward penalties (orange) and actor regularization (yellow). With reward penalties, the information flow is indirect, i.e. it is first used to update the critic model and this updated critic model is then used to update the actor model. With regularization, the information flow is directly acting on the actor model, interfacing with the environment.

This effectively forces the actor network to not only optimize the expected discounted value, but also to force its outputs corresponding to consecutive states ($\boldsymbol{s}_t$ and $\boldsymbol{s}_{t+1}$) to be similar in the chosen roughness norm (4).

Figure 3 provides a schematic overview showing the major difference between smoothness constraints imposed through the reward and those imposed through actor regularization. Notice the indirect application of the smoothness constraints on the actor model, through the critic model, when using the reward signal. As a consequence, the actor model can not be smoothed properly whenever the critic is not able to accurately capture the underlying smoothness constraints. In fact, as the critic is only an approximation of the optimal value function, there is no guarantee that the critic is able to capture this relationship at all. In practice this means many experience samples are required in order for the critic to discover the complex relationship between states, actions and accumulated returns — encoding both the performance objective and the smoothness constraints — without any certainty of success. Hence, while such an indirect information flow works reasonably well for complex functions of states and actions, such as the performance objective (1), it is needlessly complex for the applied smoothness constraints, which only depend on consecutive transitions in state-action space. In contrast, when using the regularization term, the smoothness constraints are applied directly on the actor model, without any intermediate approximation step. Moreover, in the calculation of the regularization term we can explicitly utilize the tight temporal connection in state-action space of the smoothness constraints, leading to a more sample-efficient smoothing effect.

The introduced hyperparameter $\lambda_s \geq 0$ in (6) can be tuned to trade-off performance and smoothness objectives. Low values will result in policies optimizing their future rewards, but they might be non-smooth (see the example of Section 3). High values will result in smooth policies, but might not always achieve a good performance. Remark that the effective amount of smoothing also depends on the definition of the reward signal. After all, the smoothness

weight λ_s balances the regularization term R_{smooth} *relatively* against the actor loss L_μ (6), which is proportional to the average Q-value (2). Hence, the same value of λ_s will have less smoothing impact on environments with larger rewards (in absolute value), leading to Q-values and actor loss values with higher order of magnitude, effectively suppressing the smoothness regularization term. For this reason, it is recommended to normalize the reward signal, prior to storing it in the replay buffer. Then, bounds on the L_μ term can be calculated and traded off against the maximum value of R_{smooth}, which is bounded by $\max_{i,j} \rho(\boldsymbol{a}_i, \boldsymbol{a}_j)$. Such reward normalization is also applied in the conducted experiments of Section 5 and shows the improved robustness of the smoothness weight parameter: a single λ_s value leading to acceptable policy smoothing across varying environments.

Different approximations of the used roughness metric $\bar{\rho}_\pi$ (5) and different schedules for the smoothness weight λ_s lead to different STD3 variants. In the remainder of this subsection, these different variants will be introduced. Remark that the specific combination $\text{STD3}_{\text{C,fix}}$ corresponds to the temporal smoothness regularization term of the CAPS method presented by Mysore et al. [12].

Roughness approximation The expectancy in (5) can be approximated in different ways, leading to two STD3 variants introduced below. Both approximations can reuse the same batch of sampled experience from the replay buffer $\mathcal{B}$, used to calculate (2). Hence, the extra smoothness regularization term can be easily plugged into existing training loops of off-policy actor-critic methods such as DDPG, TD3, Proximal Policy Optimization (PPO) [18] and Soft Actor Critic (SAC) [9].

The first *supervised smoothing* ($\text{STD3}_{\text{S},\bullet}$) variant uses the current action $\boldsymbol{a}_t$, as sampled from the replay buffer, and the next action $\tilde{\boldsymbol{a}}_{t+1} = \mu(\boldsymbol{s}_{t+1}; \boldsymbol{\theta}_\mu)$, as given by the policy for the next state $\boldsymbol{s}_{t+1}$, in the regularizer calculation

$$R_{\text{smooth}}(\boldsymbol{\theta}_\mu) = \mathbb{E}_{(\boldsymbol{a}_t, \boldsymbol{s}_{t+1}) \sim \mathcal{B}} \left[\rho(\boldsymbol{a}_t, \mu(\boldsymbol{s}_{t+1}; \boldsymbol{\theta}_\mu)) \right].$$

The name 'supervised' stems from the fact that the resulting smoothness regularizer forces actor network outputs to be similar to given targets (the sampled $\boldsymbol{a}_t$ actions from the replay buffer) in the chosen roughness norm (4), analogous to the classical supervised learning setting.

The second *contrastive smoothing* ($\text{STD3}_{\text{C},\bullet}$) variant uses the current action $\tilde{\boldsymbol{a}}_t = \mu(\boldsymbol{s}_t; \boldsymbol{\theta}_\mu)$ and next action $\tilde{\boldsymbol{a}}_{t+1} = \mu(\boldsymbol{s}_{t+1}; \boldsymbol{\theta}_\mu)$, both as given by the policy for the current and next states, in the regularizer calculation

$$R_{\text{smooth}}(\boldsymbol{\theta}_\mu) = \mathbb{E}_{(\boldsymbol{s}_t, \boldsymbol{s}_{t+1}) \sim \mathcal{B}} \left[\rho(\mu(\boldsymbol{s}_t; \boldsymbol{\theta}_\mu), \mu(\boldsymbol{s}_{t+1}; \boldsymbol{\theta}_\mu)) \right].$$

In this case the name 'contrastive' stems from the fact that two outputs from the same actor network are forced to be similar in the chosen roughness norm (4), comparable to the contrastive learning setting. Remark that the calculation of $\tilde{\boldsymbol{a}}_t = \mu(\boldsymbol{s}_t; \boldsymbol{\theta}_\mu)$ and the corresponding forward and backward pass through the actor network are already performed for the calculation of the actor loss L_μ (2).

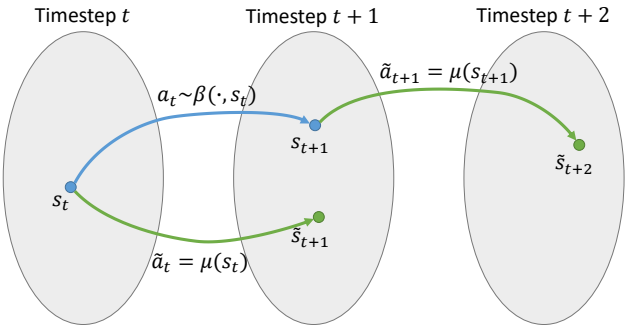


Fig. 4. Comparison of supervised and contrastive smoothing approximations. In blue are the states and actions present in the replay buffer $\mathcal{B}$. In green are the actions calculated during evaluation of the smoothness regularizer. For supervised smoothing, $\rho(\boldsymbol{a}_t, \tilde{\boldsymbol{a}}_{t+1})$ is used for the immediate roughness approximation. For contrastive smoothing, $\rho(\tilde{\boldsymbol{a}}_t, \tilde{\boldsymbol{a}}_{t+1})$ is used instead.

Hence, there is no significant computational overhead for using the contrastive variant, as compared to the supervised variant.

Notice that only a single $(\boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ or $(\boldsymbol{s}_t, \boldsymbol{s}_{t+1})$ experience sample is needed to already start improving the smoothness of the actor model for the sampled state transition. As previously mentioned this is more sample-efficient than the usage of smoothness penalties in the reward, which require multiple experience samples to uncover the underlying smoothness goal.

A comparison of the used state and action information by both variants is shown in Figure 4. The supervised variant has the strongest temporal connection between the consecutive actions $\boldsymbol{a}_t$ and $\tilde{\boldsymbol{a}}_{t+1}$ used in the regularizer calculation. More precisely, it is guaranteed that taking action $\boldsymbol{a}_t$ in state $\boldsymbol{s}_t$ can lead to state $\boldsymbol{s}_{t+1}$, where the current deterministic policy will take action $\tilde{\boldsymbol{a}}_{t+1}$. Hence, forcing the actor output $\mu(\boldsymbol{s}_{t+1}; \boldsymbol{\theta}_\mu)$ to be similar to $\boldsymbol{a}_t$ in the chosen roughness norm, will indeed improve the temporal smoothness of the policy. For the contrastive variant there is no such strong temporal connection, as taking action $\tilde{\boldsymbol{a}}_t$ in state $\boldsymbol{s}_t$ does not necessarily lead to state $\boldsymbol{s}_{t+1}$. This discrepancy (between states $\boldsymbol{s}_{t+1}$ and $\tilde{\boldsymbol{s}}_{t+1}$) will however diminish as training goes on and the behavioural policy β becomes more similar to the deterministic policy μ.

Smoothing schedules A smoothing schedule is a function $\lambda_s(f) : [0\,;1] \rightarrow [\lambda_{s,m}\,;\lambda_{s,M}]$ mapping the current training progress $f = \frac{T}{T_M}$ to a value for the smoothing weight λ_s, where the current training episode is denoted by T and the total amount of training episodes by T_M. All considered schedules are bounded, i.e., $0 \leq \lambda_{s,m} \leq \lambda_s(f) \leq \lambda_{s,M} \quad \forall f \in [0\,;1]$.

The most straightforward STD3$_{\bullet,\text{fix}}$ variant keeps the smoothness weight λ_s fixed during the whole training process. The smoothness weight then becomes another hyperparameter to tune, depending on the complexity of the environ-

ment and desired amount of policy smoothing

$$\lambda_s(f) = \lambda_s = \lambda_{s,m} = \lambda_{s,M} \quad \forall f \in [0\,;1].$$

On difficult environments, it can be hard to find good smoothness weight values under the fixed scheme. Extra smoothing constraints too early in the training process can hamper the optimization process, leading to suboptimal learned policies. The second STD3$_{\bullet,\text{lin}}$ variant tries to resolve this issue by splitting the training process in two phases: an initial phase from $f = 0$ to $f = f_{p1} < 1$, followed by a second phase until $f = 1$. In the first training phase λ_s remains equal to $\lambda_{s,m} = 0$, allowing the agent to maximally optimize the policy's performance without any smoothness constraints. Hence, during this phase, there is no difference between the smoothed and raw TD3 method. In the second training phase, after a reasonably good policy has been found, the smoothness constraints are gradually introduced by linearly increasing λ_s as training progresses (until the end of training). The resulting schedule effectively 'smoothens out' the policies obtained after the first training phase

$$\lambda_s(f) = \begin{cases} 0 & 0 \leq f \leq f_{p1} \quad \textit{(Phase 1)} \\ \lambda_{s,M} \dfrac{f - f_{p1}}{1 - f_{p1}} & f_{p1} < f \leq 1 \quad \textit{(Phase 2)} \end{cases}.$$

In practice, the determination of f_{p1} — the end of phase 1 — might require some trial-and-error experiments. Moreover, it might not always be necessary to maximally reduce the policy's roughness at the potential cost of a reduced performance. For some applications, keeping the roughness below a certain threshold may satisfy all real-world requirements on smoothness. The last STD3$_{\bullet,\text{adapt}}$ variant addresses both issues, by automatically putting more weight on smoothing or value optimization depending on a current roughness estimate $\tilde{\rho}_\pi$. This roughness estimate tries to approximate $\bar{\rho}_\pi$ (5) by averaging the measured average roughness (3) over E evaluation episodes (i.e. using the deterministic policy $\boldsymbol{a} = \mu(\boldsymbol{s}; \boldsymbol{\theta}_\mu)$ without exploration noise)

$$\tilde{\rho}_\pi = \frac{1}{E} \sum_{e=1}^{E} \frac{1}{N_e} \sum_{t=1}^{N_e} \rho(\boldsymbol{a}_{t-1}^e, \boldsymbol{a}_t^e), \tag{7}$$

where the superscript on the actions denotes the specific evaluation episode in which they occurred. This estimate is then reevaluated every T_e training episodes, keeping the smoothness weight constant in between the evaluations

$$\begin{aligned} \lambda_s(f) &= \lambda_{s,k} \quad \forall f : kT_e \leq fT_M < (k+1)T_e, \\ \lambda_{s,0} &\in [\lambda_{s,m}\,;\lambda_{s,M}], \\ \lambda_{s,k+1} &= \begin{cases} \max\{\lambda_{s,m}, s^- \lambda_{s,k}\} & \tilde{\rho}_\pi < \rho_m \\ \lambda_{s,k} & \tilde{\rho}_\pi \in [\rho_m\,;\rho_M] \\ \min\{\lambda_{s,M}, s^+ \lambda_{s,k}\} & \tilde{\rho}_\pi > \rho_M \end{cases}. \end{aligned}$$

The resulting smoothing schedule is piecewise constant and adapts the amount of smoothing throughout training, based on the specific needs. More precisely, in case the roughness estimate lies above a predefined upper threshold ρ_M, more weight is put on the policy smoothing for the next T_e training episodes — through multiplication of λ_s by $s^+ > 1$. Similarly, in case the roughness estimate lies below a predefined lower threshold ρ_m, more weight is put (again) on policy optimization for the next T_e training episodes — through multiplication of λ_s by $s^- \in (0\,;1)$.

This last variant could be seen as an ad-hoc strategy to find an approximate solution of the constrained MDP (CMDP)

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi,\tau,\sigma_0}\left[R_0\right] \\ \text{s.t. } \bar{\rho}_\pi \leq \rho_M .$$

Here, the focus is not to maximally reduce the roughness of the obtained policies (as is the case for the first two smoothing schedules), but rather to reduce the roughness of the policies to an acceptable level determined by the specified roughness thresholds. In optimal control, such a constraint is also referred to as a *slew rate constraint*.

5 Experimental results

Three different experiments were conducted to compare the different smoothed TD3 variants across different environments of varying complexity. The training and evaluation procedures are briefly described first.

For every hyperparameter configuration, the experiment is repeated five times, using five different seeds for initialization. After every T_e training episodes or k_e training timesteps, E independent evaluation episodes are executed to get an estimate of the learned policy's average performance $\tilde{R}_\pi$ and roughness $\tilde{\rho}_\pi$ (7)

$$\tilde{R}_\pi = \frac{1}{E}\sum_{e=1}^{E}\frac{1}{N_e}\sum_{t=1}^{N_e} r(\boldsymbol{s}_{t-1}^e, \boldsymbol{a}_{t-1}^e, \boldsymbol{s}_t^e).$$

To summarize these average evaluation metrics and make a comparison between different settings easier, only evaluation metrics of the best B episodes, occurring in the last T_B episodes of the training process[4] are retained. Note that the best B episodes are determined based on the average evaluation performance $\tilde{R}_\pi$ only. Combined with the five independent repeats of each experiment, this leads to $5B$ datapoints, from which comparison statistics are calculated (e.g. a mean value and standard deviation).

[4] This is to guarantee proper convergence of both the performance and smoothness objective on every environment.

5.1 Highway overtaking

As the first experiment, let us quickly revisit the motivational example of Section 3, where the goal was to learn a policy that can overtake a slow vehicle in the rightmost lane of a highway (Figure 1). First, five policies (initialized using different seeds) were trained using the TD3 method without smoothness constraints. Afterwards, five policies (reusing the same five seeds) were trained using the $\text{STD3}_{\text{S,lin}}$ method to smoothen out the previously obtained policies after an initial 200 training episodes ($f_{p1} = 2/3$). While almost all TD3 policies suffered from jerky actions throughout training, the smoothness has improved a lot for the STD3 policies. Only one policy still had some oscillatory reference actions after smoothing, but for only a fraction of the time as compared to the unsmoothed policies. Figure 2 shows an evaluation episode of one of the five policies after 250 training episodes (i.e. after 50 smoothing episodes for the STD3 policy).

5.2 OpenAI benchmarks

In this second experiment, the different STD3 variants will be compared against each other (and the standard TD3 method) on 10 commonly used OpenAI gym environments[5]. We use a customized version of the latest Stable-Baselines implementation[6] to perform these experiments. Their tuned hyperparameters for the TD3 algorithm are reused with a few exceptions for some environments requiring a longer training time. A summary of the used hyperparameters and full environment names can be found in Appendix B. The used smoothness parameters are summarized in Table 1. Notice that for the fixed and linear smoothing schedule, the same parameters could be reused across all environments. This was possible due to the normalization of states and rewards, prior to storage in the replay buffer, and shows the robustness of these smoothness parameters. For the adaptive smoothing schedule, the extra imposed smoothness constraint was set as to reduce the roughness of the policies by half, as compared to standard TD3. More precisely, we put the maximum threshold ρ_M equal to approximately half the roughness of policies obtained using default TD3. The minimum threshold was set to roughly 90% of the maximum threshold's value.

The training and evaluation procedure outlined at the beginning of this section was followed using five independent repeats with $E = 5$ evaluation episodes every $k_e = 5000$ training timesteps. Evaluation metrics of the five best episodes occurring in the last 20% of the training process were used to make the comparison between different configurations ($B = 5$, $T_B = 0.2T_M$). The mean values and standard deviations for the performance and roughness metrics are summarized in Table 2.

In general, all investigated STD3 variants significantly improve the policy smoothness. However, some do so at the cost of a reduced performance. For example, the $\text{STD3}_{\text{S,fix}}$ method consistently leads to the smoothest policies, but

[5] https://gym.openai.com
[6] https://github.com/DLR-RM/stable-baselines3

Table 1. Smoothness parameters for the different environments. The normalized roughness norm ρ_{norm} was used for both the regularizer calculation R_{smooth} and roughness estimation $\tilde{\rho}_\pi$. The remaining parameters for the adaptive variant were set as follows for every environment: $\lambda_{s,0} = 1 \cdot 10^{-4}$, $\lambda_{s,m} = 1 \cdot 10^{-6}$, $\lambda_{s,M} = 1$.

Environment	**STD3$_{\bullet,\text{fix}}$**	**STD3$_{\bullet,\text{lin}}$**		**STD3$_{\bullet,\text{adapt}}$**	
	λ_s	f_{p1}	$\lambda_{s,M}$	ρ_m	ρ_M
Ant	0.2	0.6	0.4	0.36	0.4
Bipedal	0.2	0.6	0.4	0.16	0.18
Hopper	0.2	0.6	0.4	0.04	0.05
IDP	0.2	0.6	0.4	0.08	0.09
IPS	0.2	0.6	0.4	0.08	0.1
Lunar	0.2	0.6	0.4	0.18	0.2
Minitaur	0.2	0.6	0.4	0.36	0.4
MCC	0.2	0.6	0.4	0.004	0.005
Pendulum	0.2	0.6	0.4	0.08	0.1
Walker	0.2	0.6	0.4	0.16	0.18

often not to the best performing ones. The reverse situation is also observable for the STD3$_{\text{C,fix}}$ method: this method leads to the best performing policies, but other methods can typically reduce the roughness slightly more. It should be noted however that results lie close together for some environments. Furthermore, the addition of extra smoothness constraints does not always lead to a reduction in performance. On five environments the best performing TD3 policies are outperformed by an STD3 variant. In particular, the STD3$_{\text{C,lin}}$ method seems to find the best balance between performance and smoothness, as it leads most often (on five environments) to both the best performing and smoothest policies.

It might not always be required to obtain the absolute best performing or smoothest policy though. Depending on the performance-smoothness trade-off acceptable for a given application, the best STD3 variant can be selected. The maximum flexibility in defining the desired smoothing behaviour is obtained using the adaptive smoothing schedules. As can be seen from Table 2, reducing the roughness by 50% seems to succeed in at least 6 different environments. This comes without large performance costs, as we still obtain best performing policies in half of the environments. The adaptive smoothing schedules seem to have the most difficulty on environments where the best performing TD3 policies have a high smoothness variability (high roughness variance in Table 2). A possible explanation for this behaviour is the observed abrupt vanishing and reappearance of rough actions throughout training. This might lower the smoothness weight, even though the policies are still 'vulnerable' to emerging jerky actions. One possible solution for this, is the usage of an exponential moving average for the roughness estimate, instead of recalculating it from scratch every evaluation. In

Table 2. Benchmark results of the different STD3 variants. All values are relative changes with respect to the mean value obtained by the TD3 algorithm. The best performing policies (top) and smoothest policies (bottom) are highlighted in bold. The bottom row summarizes the data, denoting the number of environments for which *both* the best performing policy *and* smoothest policy was obtained using the given method. For the adaptive variants, the number in parentheses is the amount of environments for which the best performing policy was obtained, under the given smoothness constraint.

Environment	**Average return (↑) [%]**						
	TD3	$\text{STD3}_{\text{S,fix}}$	$\text{STD3}_{\text{S,lin}}$	$\text{STD3}_{\text{S,adapt}}$	$\text{STD3}_{\text{C,fix}}$	$\text{STD3}_{\text{C,lin}}$	$\text{STD3}_{\text{C,adapt}}$
Ant	$\mathbf{0.00 \pm 4.92}$	-21.87 ± 5.33	-11.47 ± 7.29	-20.26 ± 38.03	$\mathbf{-4.00 \pm 8.68}$	$\mathbf{-1.62 \pm 6.29}$	$\mathbf{-1.19 \pm 3.62}$
Bipedal	0.00 ± 0.88	-3.45 ± 1.02	-0.24 ± 1.24	-0.09 ± 2.09	$\mathbf{2.95 \pm 0.61}$	$\mathbf{2.99 \pm 0.56}$	1.74 ± 0.39
Hopper	0.00 ± 4.18	-4.92 ± 5.76	-0.86 ± 4.47	-2.25 ± 3.50	$\mathbf{9.70 \pm 2.60}$	-2.48 ± 4.11	2.72 ± 5.74
IDP	$\mathbf{0.00 \pm 0.02}$	-0.06 ± 0.01	-0.03 ± 0.02	$\mathbf{-0.02 \pm 0.04}$	-0.02 ± 0.03	$\mathbf{-0.01 \pm 0.02}$	$\mathbf{-0.02 \pm 0.02}$
IPS	$\mathbf{0.00 \pm 0.09}$	-2.58 ± 2.65	-0.95 ± 0.46	$\mathbf{-0.04 \pm 0.07}$	$\mathbf{-0.08 \pm 0.21}$	-0.19 ± 0.24	-0.15 ± 0.17
Lunar	0.00 ± 2.88	$\mathbf{5.02 \pm 1.75}$	1.60 ± 4.49	1.51 ± 2.10	$\mathbf{5.91 \pm 4.24}$	1.94 ± 3.11	2.30 ± 2.70
Minitaur	0.00 ± 15.42	-27.89 ± 32.24	-23.32 ± 22.35	3.15 ± 27.48	$\mathbf{25.84 \pm 9.73}$	-3.03 ± 26.09	$\mathbf{19.75 \pm 6.22}$
MCC	0.00 ± 0.15	-19.21 ± 40.61	$\mathbf{0.72 \pm 0.23}$	0.24 ± 0.11	0.36 ± 0.10	0.35 ± 0.09	-20.10 ± 40.00
Pendulum	$\mathbf{0.00 \pm 26.96}$	$\mathbf{-3.35 \pm 28.93}$	$\mathbf{-4.37 \pm 28.59}$	$\mathbf{-1.40 \pm 29.14}$	$\mathbf{-1.52 \pm 26.10}$	$\mathbf{-0.36 \pm 26.57}$	$\mathbf{-2.31 \pm 26.73}$
Walker	$\mathbf{0.00 \pm 4.55}$	$\mathbf{-2.16 \pm 2.94}$	$\mathbf{1.02 \pm 3.21}$	$\mathbf{-1.60 \pm 6.16}$	$\mathbf{2.18 \pm 5.32}$	$\mathbf{-1.78 \pm 3.80}$	$\mathbf{-1.82 \pm 3.44}$
Environment	**Average roughness (↓) [%]**						
	TD3	$\text{STD3}_{\text{S,fix}}$	$\text{STD3}_{\text{S,lin}}$	$\text{STD3}_{\text{S,adapt}}$	$\text{STD3}_{\text{C,fix}}$	$\text{STD3}_{\text{C,lin}}$	$\text{STD3}_{\text{C,adapt}}$
Ant	0.00 ± 9.89	-83.93 ± 1.02	-83.46 ± 1.97	-63.53 ± 18.81	-84.62 ± 1.91	$\mathbf{-87.04 \pm 1.65}$	-55.17 ± 5.64
Bipedal	0.00 ± 7.61	$\mathbf{-85.31 \pm 2.40}$	$\mathbf{-84.45 \pm 3.05}$	-48.76 ± 4.46	-82.08 ± 4.78	$\mathbf{-85.59 \pm 2.54}$	-53.15 ± 5.64
Hopper	0.00 ± 27.74	$\mathbf{-70.23 \pm 4.62}$	-65.40 ± 8.73	-43.37 ± 8.60	-66.12 ± 8.95	$\mathbf{-75.98 \pm 8.93}$	-48.31 ± 7.49
IDP	0.00 ± 58.45	$\mathbf{-97.82 \pm 0.69}$	$\mathbf{-97.31 \pm 1.10}$	-16.07 ± 88.20	$\mathbf{-98.66 \pm 2.29}$	$\mathbf{-98.92 \pm 1.70}$	-33.76 ± 42.98
IPS	0.00 ± 146.19	$\mathbf{-99.70 \pm 0.22}$	-98.61 ± 0.64	-0.27 ± 139.53	$\mathbf{-99.74 \pm 0.20}$	-99.38 ± 0.40	-68.86 ± 36.79
Lunar	0.00 ± 32.70	$\mathbf{-96.79 \pm 1.11}$	$\mathbf{-96.96 \pm 1.26}$	-48.66 ± 18.96	-91.13 ± 2.44	-92.04 ± 3.19	-48.00 ± 25.37
Minitaur	0.00 ± 16.36	$\mathbf{-82.00 \pm 5.70}$	-76.02 ± 6.79	-49.75 ± 4.14	-59.03 ± 4.63	-66.41 ± 11.81	-52.63 ± 4.38
MCC	0.00 ± 63.12	$\mathbf{-74.66 \pm 14.05}$	$\mathbf{-71.53 \pm 5.11}$	-33.44 ± 24.95	$\mathbf{-68.87 \pm 2.37}$	$\mathbf{-69.75 \pm 4.27}$	-20.68 ± 69.77
Pendulum	0.00 ± 114.96	$\mathbf{-98.81 \pm 0.51}$	$\mathbf{-98.75 \pm 0.51}$	-38.01 ± 82.90	-98.17 ± 1.22	$\mathbf{-98.74 \pm 0.74}$	13.20 ± 125.25
Walker	0.00 ± 18.60	-77.15 ± 2.55	-77.42 ± 3.22	-48.39 ± 9.01	-76.94 ± 4.24	$\mathbf{-82.79 \pm 4.38}$	-52.17 ± 11.10
# Best	–	2	2	– (4)	1	**5**	– (5)

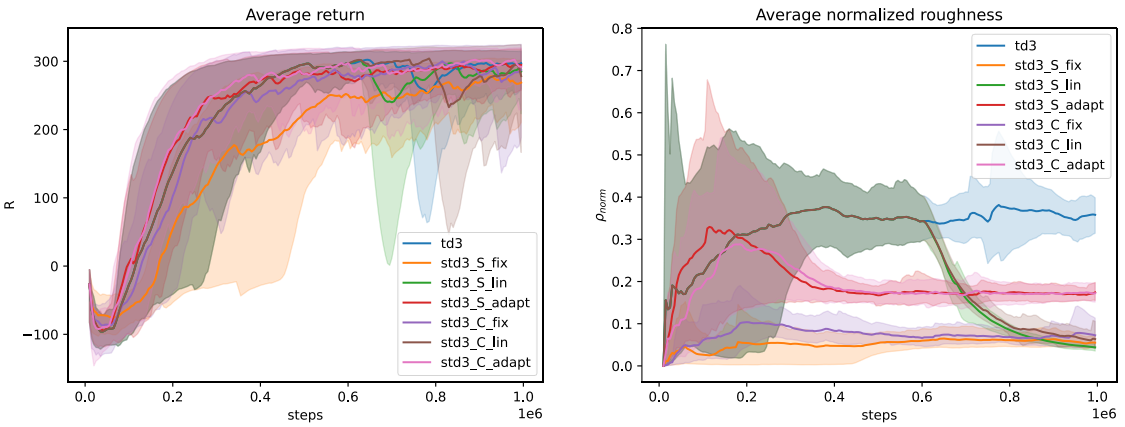


Fig. 5. Exponentially smoothed ($\alpha = 0.1$) evolution of the average return (left) and roughness (right) of the policies as training goes on for the Bipedal environment.

this way, past non-smooth behaviour is accounted for longer. It is left for future work to investigate such more elaborate smoothing schemes.

Finally, Figure 5 shows the different smoothing schedules' effect on the policy's roughness throughout training. The simplest fixed scheme immediately acts on the policy, from the very start of the training process, giving no chance for policies to become too rough. In the linear scheme, the same roughness behaviour as for the TD3 method is obtained during the initial training phase. As soon as the smoothing phase starts, the roughness is drastically reduced to roughly the same level as the fixed scheme. Both of these clearly try to reduce the roughness as much as possible. The final adaptive scheme starts to smoothen out the policies as soon as the predefined threshold is crossed, after which the roughness settles around this threshold value.

5.3 Highway driving

In this last experiment, we investigate the best performing $\text{STD3}_{\text{C,lin}}$ variant and compare it against its supervised counterpart on a more complex environment. The same simulator as in the first experiment is used, but this time there are multiple moving vehicles on the three-lane highway. The objective in this environment is to travel as fast as possible, while respecting all traffic rules (speed limit, keep right) and safety constraints (preventing crashes). Details can be found in Appendix A.

Once again, all configurations are repeated five times with differently seeded initializations. Average performance and smoothness metrics are calculated from one evaluation episode after every training episode ($E = T_e = 1$). The smoothness estimate is calculated as

$$\tilde{s}_\pi = \frac{1}{N_e} \sum_{t=1}^{N_e} \big(\exp[-\rho(\boldsymbol{a}_{t-1}, \boldsymbol{a}_t)] - 1 \big),$$

giving values closer to 0 for smoother policies and values closer to -1 for non-smooth policies. The performance metric is the accumulated sum of normalized rewards *without smoothness penalty* r_S, with maximum value 0 and minimum value -1. All policies were trained for $T_M = 300$ episodes with $k_M = 5000$ timesteps, smoothing started after one third of the training was done ($f_{p1} = 1/3$ empirically determined) using different values of the final (maximum) smoothing weight $\lambda_{s,M}$ and of the smoothness penalty weight w_S in the reward. Performance and smoothness statistics are calculated from the best 20 episodes occurring in the second half of the training process ($B = 20$, $T_B = 0.5T_M$). A summary of the results is shown in Figure 6. The first experiment (on top) compares the $\text{STD3}_{\text{S,lin}}$ method with the standard TD3 method using smoothness penalties in the reward. Clearly, the smoothness of the obtained policies is increased for both approaches. However, using smoothness penalties quickly becomes impractical, as performance starts to deteriorate for increasing values of w_S. Using smoothed TD3 on the other hand, results in policies having higher smoothness values without any performance reduction. Naturally, this only holds up to certain

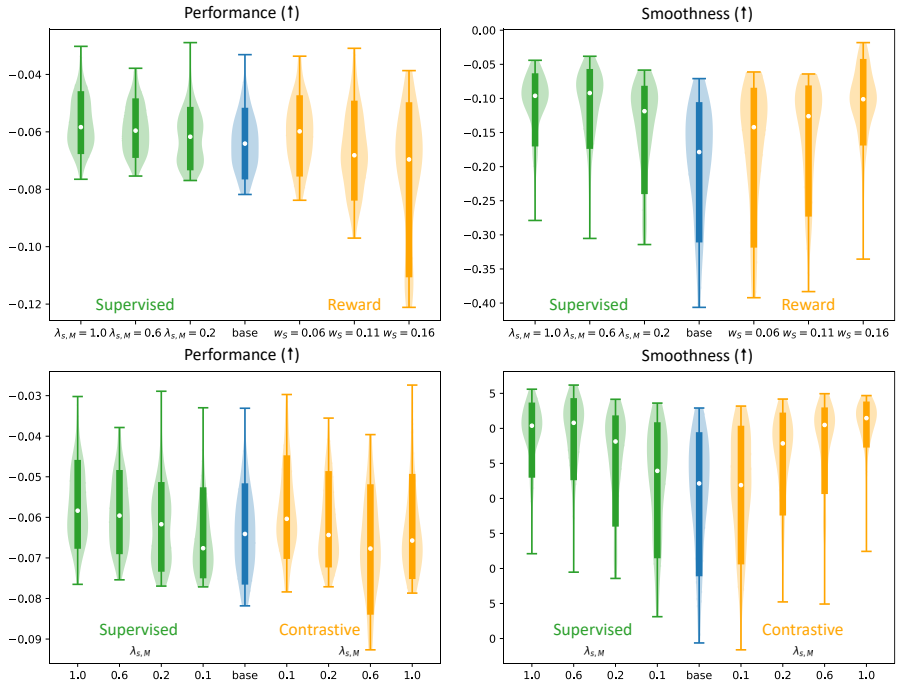


Fig. 6. Comparison of performance (left) and smoothness (right) for policies trained on the highway driving environment. In blue the standard TD3 method without any smoothness constraints or penalties. In green the supervised $\text{STD3}_{\text{S,lin}}$ method. In orange the policies trained with extra smoothness penalties in the reward signal (top) or using the contrastive $\text{STD3}_{\text{C,lin}}$ method (bottom). The whiskers denote the minimum/maximum values, the shaded area shows an estimate of the underlying distribution, the middle rectangle spans from the first to the third quartile and the white dot shows the mean value.

limits but policies trained using STD3 were found to be much more robust to such performance declines empirically. Hence less time can be spent on finetuning the trade-off between performance and smoothness, which typically required trying to fit in smoothness penalties into an already existing reward signal.

In the second experiment (bottom of Figure 6), the constrastive and supervised STD3 variants were compared (both using the 'lin' smoothing schedule). Both lead to roughly the same amount of smoothness improvement for different values of $\lambda_{s,M}$. Performance stays roughly at the same level, although there is a slight increase for the supervised variant and a slight decrease for the contrastive variant. This might be a bit surprising, as the results on the openAI gym environments seemed to indicate the contrastive variants had superior performance. But this confirms the fact that different environments require different smoothing measures. For the simplest environments, an extra penalty in the reward might suffice. As complexity increases, the smoothed TD3 variants become necessary to prevent severe performance deterioration. Finally, for the most complex environments (such as chaotic systems [2]), it seems the stronger temporal connection of actions in the supervised smoothing setting, makes them more relevant. In such environments initial policy estimates might be far off from the later, more optimal policies; and slight changes in the chosen actions could lead to vastly different state transitions. Both contributing to higher discrepancies in the compared states of the contrastive smoothing method (see Figure 4).

6 Conclusion

In this paper we introduced different smoothed TD3 (STD3) variants to improve the learned policy's temporal smoothness. The specific choice of roughness metric (5) used for the calculation of both the smoothness regularization term and the smoothness estimate, makes it easily combinable with existing off-policy, policy-based and actor-critic reinforcement learning algorithms. Experiments using normalized returns and roughness metrics show that the extra smoothness weight hyperparameter generalizes well across a variety of different environments, leading to smooth policies without significant performance deterioration. For more fine-grained control over the desired smoothness–performance trade-off, a proper smoothing schedule can be selected. From these schedules, the adaptive smoothing variant is the most versatile. Using an estimate of the currently learned policy's roughness on evaluation episodes, it tries to automatically reduce this policy's roughness below a predefined threshold set at the start of training. The resulting policy is an approximate solution of the constrained MDP with added smoothness constraints.

A possible path forward is the application of the introduced smoothing regularizers to other actor-critic methods, such as PPO and SAC. Although a similar investigation by Mysore et al. [12] observed smoothness regularization to be mostly effective for TD3 as "*soft-policies such as PPO and SAC appear to learn relatively smoother policies on their own*". Another direction of future work can be the investigation of other methods to deal with constrained MDPs, such as

Constrained Policy Optimization (CPO) [1] or Lagrangian methods [20], and compare them with the adaptive STD3 variant introduced here.

Acknowledgements

The presented results were obtained under Ford Alliance Project KUL0076, funded by Ford.

The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

Bibliography

[1] Achiam, J., Held, D., Tamar, A., Abbeel, P.: Constrained policy optimization. In: 34th International Conference on Machine Learning, ICML 2017. vol. 1, pp. 30–47 (2017)

[2] Bucci, M.A., Semeraro, O., Allauzen, A., Wisniewski, G., Cordier, L., Mathelin, L.: Control of chaotic systems by deep reinforcement learning. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **475**(2231), 20190351 (2019). https://doi.org/10.1098/rspa.2019.0351

[3] Chen, W., Xiao, H., Wang, Q., Zhao, L., Zhu, M.: Lateral Vehicle Dynamics and Control. John Wiley & Sons, Ltd, 2nd edn. (2016). https://doi.org/10.1002/9781118380000

[4] Chisari, E., Liniger, A., Rupenyan, A., Van Gool, L., Lygeros, J.: Learning from Simulation, Racing in Reality. arXiv preprint 2011.13332 [cs.RO] (2020)

[5] Fujimoto, S., Van Hoof, H., Meger, D.: Addressing Function Approximation Error in Actor-Critic Methods. In: 35th International Conference on Machine Learning, ICML 2018. vol. 4, pp. 2587–2601 (2018)

[6] Girosi, F., Jones, M., Poggio, T.: Regularization Theory and Neural Networks Architectures. Neural Computation **7**(2), 219–269 (1995). https://doi.org/10.1162/neco.1995.7.2.219

[7] Green, P., Silverman, B.W.: Nonparametric Regression and Generalized Linear Models. Chapman and Hall/CRC (may 1993). https://doi.org/10.1201/b15710

[8] Ha, S., Xu, P., Tan, Z., Levine, S., Tan, J.: Learning to Walk in the Real World with Minimal Human Effort. arXiv preprint 2002.08550 [cs.RO] (2020)

[9] Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: 35th International Conference on Machine Learning, ICML 2018. vol. 5, pp. 2976–2989 (2018)

[10] Kesting, A., Treiber, M., Helbing, D.: General lane-changing model MOBIL for car-following models. Transportation Research Record pp. 86–94 (2007). https://doi.org/10.3141/1999-10

[11] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. In: 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings (2016)

[12] Mysore, S., Mabsout, B., Mancuso, R., Saenko, K.: Regularizing Action Policies for Smooth Control with Reinforcement Learning. arXiv preprint 2012.06644 [cs.RO] (2020)

[13] Nageshrao, S., Tseng, H.E., Filev, D.: Autonomous highway driving using deep reinforcement learning. Conference Proceedings - IEEE International

Conference on Systems, Man and Cybernetics pp. 2326–2331 (mar 2019). https://doi.org/10.1109/SMC.2019.8914621
[14] Raffin, A., Kober, J., Stulp, F.: Smooth Exploration for Robotic Reinforcement Learning. arXiv preprint 2005.05719 [cs.LG] (2020)
[15] Rodriguez-Ramos, A., Sampedro, C., Bavle, H., de la Puente, P., Campoy, P.: A Deep Reinforcement Learning Strategy for UAV Autonomous Landing on a Moving Platform. Journal of Intelligent and Robotic Systems: Theory and Applications **93**(1-2), 351–366 (2019). https://doi.org/10.1007/s10846-018-0891-8
[16] Rosca, M., Weber, T., Gretton, A., Mohamed, S.: A case for new neural network smoothness constraints. arXiv preprint 2012.07969 [stat.ML] (2020)
[17] Saxena, D.M., Bae, S., Nakhaei, A., Fujimura, K., Likhachev, M.: Driving in Dense Traffic with Model-Free Reinforcement Learning. In: Proceedings - IEEE International Conference on Robotics and Automation. pp. 5385–5392 (sep 2020). https://doi.org/10.1109/ICRA40945.2020.9197132
[18] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal Policy Optimization Algorithms. arXiv preprint 1707.06347 [cs.LG] (2017)
[19] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms. In: 31st International Conference on Machine Learning, ICML 2014. vol. 1, pp. 605–619 (2014)
[20] Stooke, A., Achiam, J., Abbeel, P.: Responsive safety in reinforcement learning by PID Lagrangian Methods. arXiv preprint 2007.03964 [math.OC] pp. 9070–9080 (2020)
[21] Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, A Bradford book, vol. 258. MIT Press, 1st edn. (1998)
[22] Treiber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **62**(2), 1805–1824 (2000). https://doi.org/10.1103/PhysRevE.62.1805
[23] Wang, P., Chan, C.Y., De La Fortelle, A.: A Reinforcement Learning Based Approach for Automated Lane Change Maneuvers. In: IEEE Intelligent Vehicles Symposium, Proceedings. vol. 2018-June, pp. 1379–1384 (oct 2018). https://doi.org/10.1109/IVS.2018.8500556

A Autonomous highway driving environment

The results shown in Section 5 for the highway overtaking and driving environments are obtained using a proprietary highway simulator. In this section the most relevant components of this simulator will be briefly discussed. See also Figure 1 for a schematic overview of the overtaking environment.

A.1 Roads

All experiments were conducted on a three lane highway. For the overtaking environment, this highway was straight along the whole trajectory. For the driving environment, the highway was a closed-loop circuit, with both straight and curved segments. The maximum speed limit was set to 30m/s in all lanes, although some vehicles were instructed to slightly deviate from this limit, to get more varying situations on the road.

A.2 Vehicles

Every vehicle in the simulator follows the kinematic bicycle model (KBM) [3] to update its state based on the selected inputs

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\psi+\beta) \\ v\sin(\psi+\beta) \\ \frac{v}{l_r}\sin\beta \\ \frac{a}{\cos\beta} \end{bmatrix} \quad \beta = \arctan\left(\frac{l_r}{l_f+l_r}\tan\delta\right).$$

The vehicle's local state vector consists of an absolute x and y position, a heading angle ψ and velocity v. The vehicle can be controlled through its inputs, consisting of a steering angle δ and a longitudinal acceleration a. To make the control task of the virtual driver (agent) easier, extra low level controllers are used to stabilize the vehicle on the road, allowing the agent to select high-level steering actions $\boldsymbol{a}$, consisting of a desired longitudinal velocity and desired lateral position, to solve the driving task. To take correct high level steering decisions, the virtual driver needs some extra information about other traffic participants in its neighbourhood. This information is all gathered in the agent's observation vector $\boldsymbol{s}$, containing local information such as the vehicle's offset w.r.t. different lane centers and its velocity components; and relative information such as relative gaps and velocities w.r.t. neighbouring traffic. Internally, the simulator discretizes time with step size $\Delta t = 0.1s$ and a Runge-Kutta integration scheme to calculate subsequent states.

A.3 Policies

Every vehicle is controlled by a policy, mapping observations $\boldsymbol{s}$ to suitable high-level actions $\boldsymbol{a}$. The policy of the autonomous vehicle is learned using any of the described RL methods in this paper. The policies of the other vehicles in the

simulation environment are fixed beforehand. In the overtaking environment, the policy used for the slow leading vehicle (blue in Figure 1) yields the same, fixed actions for every state, keeping the vehicle within the initial lane at a constant velocity. In the driving environment, a mixture of vehicles equipped with a custom rule-based policy and a policy implementing the 'Intelligent Driver Model' (IDM) [22] and 'Minimizing Overall Braking Induced by Lane change' (MOBIL) [10] is used. Both policies try to mimick rudimentary human driving behaviour, although being fully deterministic. Safety of the chosen actions was guaranteed through an extra safety check, similar to what is done by Nageshrao et al. [13]. Unsafe actions are mapped to the nearest safe actions, before being passed to the lower level controllers, avoiding most collisions.

A.4 Reward

The used reward signal is calculated as a weighted sum of different penalties

$$r = w_F r_F + w_V r_V + w_C r_C + w_R r_R + w_B r_B + w_S r_S.$$

The first 'frontal' component r_F gives a penalty whenever the following distance to the leading vehicle is smaller than a predefined threshold. The 'velocity' component r_V gives a penalty whenever the virtual driver is not travelling at or near the maximum allowed speed. The third 'center' component r_C gives a penalty whenever the vehicle is not correctly aligned within its current lane – travelling central in the lane. To force the virtual driver to keep right whenever possible, the 'right' penalty r_R is given whenever there is a free lane to the right available. Finally, for some experiments a penalty for non-smooth policies is given in the reward through the r_S component.

The final reward is rescaled by the sum of all composing weights, such that it always lies in the interval $[-1\,;0]$.

B TD3 hyperparameters for the gym environments

The table below shows the used hyperparameters for the TD3 algorithm (and its smoothed variants) on the 10 used OpenAI gym environments used in the experiments section. Most of these values correspond to the tuned hyperparameters of the Stable-Baselines3 repository[7], the differences are highlighted in bold.

[7] https://github.com/DLR-RM/rl-baselines3-zoo/blob/master/hyperparams/td3.yml

Table 3. Overview of the used hyperparameters for each environment. The shown hyperparameters are: maximum timesteps per episode k_M, total training timesteps $k_M \cdot T_M$, distribution of the exploration noise $\epsilon \sim E$, discount factor γ, replay buffer size $|\mathcal{B}|$. The exploration noise generators are: the normal distribution $N(\mu, \sigma)$ with mean μ and standard deviation σ, the Ornstein-Uhlenbeck process $O(\mu, \sigma, \theta)$ with mean μ, standard deviation σ and damping θ.

Environment	k_M	$k_M T_M$	E	γ	$\lvert\mathcal{B}\rvert$
Ant (AntBulletEnv-v0)	1000	$1 \cdot 10^6$	$N(0, 0.1)$	0.98	$2 \cdot 10^5$
Bipedal (BipedalWalker-v3)	1600	$1 \cdot 10^6$	$N(0, 0.1)$	0.98	$2 \cdot 10^5$
Hopper (HopperBulletEnv-v0)	1000	$1 \cdot 10^6$	$N(0, 0.1)$	0.98	$2 \cdot 10^5$
IDP (InvertedDoublePendulumBulletEnv-v0)	1000	$1 \cdot 10^6$	$N(0, 0.1)$	0.98	$2 \cdot 10^5$
IPS (InvertedPendulumSwingupBulletEnv-v0)	1000	$\mathbf{5 \cdot 10^5}$	$N(0, 0.1)$	0.98	$2 \cdot 10^5$
Lunar (LunarLanderContinuous-v2)	1000	$\mathbf{1 \cdot 10^6}$	$N(0, 0.1)$	0.98	$2 \cdot 10^5$
Minitaur (MinitaurBulletEnv-v0)	1000	$1 \cdot 10^6$	$N(0, 0.1)$	0.99	$1 \cdot 10^6$
MCC (MountainCarContinuous-v0)	999	$\mathbf{5 \cdot 10^5}$	$O(0, 0.5, 0.15)$	0.99	$1 \cdot 10^6$
Pendulum (Pendulum-v0)	200	$\mathbf{1 \cdot 10^5}$	$N(0, 0.1)$	0.98	$2 \cdot 10^5$
Walker (Walker2DBulletEnv-v0)	1000	$1 \cdot 10^6$	$N(0, 0.1)$	0.98	$2 \cdot 10^5$

Table 4. Overview of the used hyperparameters, common across all used environments.

Common hyperparameters		
Learning rate (actor + critic)	η	$1 \cdot 10^{-3}$
Warmup timesteps		10000
Batch size	B	100
Policy update delay	d	2
Target policy noise distribution		$N(0, 0.2)$
Target policy noise clipping		$[-0.5\,;0.5]$
Polyak averaging constant	τ	$5 \cdot 10^{-3}$
Network architecture – hidden dimensions (actor + critic)		400×300

Explainable AI through the Learning of Arguments

Jonas Bei, David Pomerenke, Lukas Schreiner, Sepideh Sharbaf, Pieter Collins, and Nico Roos[1]

Data Science and Knowledge Engineering, Maastricht University,
Maastricht, The Netherlands
{jy.bei,d.pomerenke,lj.schreiner,s.sharbaf}@student.maastrichtuniversity.nl,
{pieter.collins,roos}@maastrichtuniversity.nl
http://www.maastrichtuniversity.nl/dke/

Abstract. *Learning arguments* is highly relevant to the field of explainable artificial intelligence. It is a family of symbolic machine learning techniques that is particularly human-interpretable. These techniques learn a set of arguments as an intermediate representation. Arguments are small rules with exceptions that can be chained to larger arguments for making predictions or decisions.
We investigate the learning of arguments, specifically the learning of arguments from a 'case model' proposed by Verheij [34]. The case model in Verheij's approach are cases or scenarios in a legal setting. The number of cases in a case model are relatively low. Here, we investigate whether Verheij's approach can be used for learning arguments from other types of data sets with a much larger number of instances. We compare the learning of arguments from a case model with the HeRO algorithm [15] and learning a decision tree.

Keywords: Explainable AI · Argumentation · Learning Arguments · Data Mining

1 Introduction

Explainable AI Artificial intelligence, in a societal context, is confronted with a variety of requirements that are recently being investigated by the research fields around explainable, responsible and socially aware artificial intelligence [31]. Here, we are concerned with explainability, that is, making the criteria transparent that underlie the decision of an algorithm.

Explainability is also increasingly becoming a *legal* requirement of algorithms. In many countries, which as of recently includes the Netherlands [25, 27], administrative and judicative decisions that have been supported by an algorithm are required to be comprehensible for judges and citizens [9]. The General Data Protection Regulation of the EU (see [12]), as well as similar legislation in the United States gives citizens a right to explainability also towards companies; albeit only when important decisions such as credit status are involved.

Surveys have been undertaken as to which machine learning techniques are suitable for explainable artificial intelligence, according to a range of sub-criteria. The result is

[1] The authors thank Julien Havel for his contribution to the initial phase of the reported research.

that decision trees and approaches based on deductive logic are the most suitable techniques [3, 36]. Here we investigate the learning of arguments, which can be classified broadly as a deductive logic approach.

Benefits of arguments Arguments provide reasons for believing conclusions given data [33]. Providing arguments for conclusions, considering exceptions to these arguments, and putting multiple small arguments together to build larger, convincing arguments, is the human way of justifying things. Learning of arguments from data sets and using these arguments for future decision making will provide more transparency than black box approaches. This transparency is important in domains such as law, public administration, health care, etc, as well as to the discovery of scientific explanations.

Decision making based on learned argument addresses three problems:
The first problem is the mentioned requirement of the explainability of the decision of the algorithm. An algorithm that substantiates its claims with arguments can, if the arguments are properly presented, be understood by a human. Thus, humans can detect potential errors in the algorithm's decision, or, hopefully, verify that no such errors have been made. This increases trust between human and machine [9].

The second problem is that experts (or even non-expert humans) may possess some relevant knowledge that can improve learning form training data, such as known causal relationships between some of the attributes of the data. Machine learning systems that produce arguments can incorporate the knowledge of both the data set and the expert.

The third problem is that humans may pose certain requirements towards the justification of a decision that are in conflict with the training data. Important example are racial, sexual, and other biases, that may be present in the training data, and would lead to the perpetuation of discrimination (and hence, further biased data sets) in the future. In order to avoid vicious circles of discrimination, humans may wish to reject discriminatory decisions implied by the data. This may also be realized by discarding, for example, racially motivated arguments.

Research aims Verheij [34] proposed an approach for learning arguments from a 'case model'. The case model in Verheij's approach are cases or scenarios in a legal setting, and the number of cases in a case model are relatively low. We investigate whether Verheij's approach can be used for learning arguments from other types of data sets with a much larger number of instances. We compare the learning of arguments from a case model with another approach for learning arguments, the HeRO algorithm [15], and with learning a decision tree.

Paper outline The next section describes the related work. Section 3 describes the preliminaries and Section 4 describes our implementation of Verheij's approach [34] as well as the other approaches that we implemented for comparison. Section 5 describes experimental evaluation and Section 6 concludes the paper.

2 Related work

Here we give a concise overview on the most relevant related work.

Argumentation The modern view of argumentation was introduced by Toulmin [33]. He describes an argument as a (defeasible) warrant for a claim / conclusion given some data / premises. One of the first argumentation systems based on this idea was developed by Pollock [24], who extended predicate logic with defeasible and undercutting rules. An important issue in argumentation systems that make use of defeasible information, is determining which arguments are valid. Dung [10] showed that this problem can be described by an *argumentation framework*, which is a couple consisting of a set of atomic arguments with an attack relation over the arguments. He defines three argumentation semantics for determining the set of valid arguments given an argumentation framework, namely, the grounded, stable and preferred semantics. Arguments learned by Verheij's approach can be evaluated using the grounded semantics, while arguments learned by the HeRo algorithm may require the preferred semantics.

Learning arguments Kakas and Michael [18] give an insightful overview on argumentation in machine learning, enumerating multiple use cases of arguments. Here, we are concerned with argumentation as the target language for learning. Within this use case, they distinguish two paradigms. In the first paradigm, arguments are potentially large monolithic rules that directly map input facts to output facts. This paradigm comprises decision lists, exception lists, inductive logic programming with exceptions, and random forest methods. In the second paradigm, arguments consist of multiple chained smaller arguments, with intermediate concepts connecting the arguments. The smaller arguments describe local relations, that is, relations that only involve a small number of attributes. Within this paradigm fall the *NERD* algorithm [20], *machine coaching*, and *SLAP*.

Two algorithms are explicitly concerned with the mining of defeasible rules: Firstly, the *DefGen* algorithm uses association rule mining, for which highly optimized algorithms for big data exist, and post-processes the output by applying relevance criteria [13]. This high-level structure can also be found in our *Pruned Search* algorithm introduced in Section 4.2. Secondly, the *HeRO* algorithm iteratively applies the criterion of *information gain*, taking inspiration from decision list mining and covering rule algorithms. We have implemented the HeRO algorithm; see Subsection 4.2.

Other rule-based learning approaches Competing approaches for the explainable learning of rules are decision trees, relational learning and inductive logic programming, and probabilistic and causal networks.

While decision trees are equivalent to sets of classification rules [37, ch. 3.4], [14, p. 358], the rules to which they correspond are long and unstructured. Domain experts prefer to work with well-structured sets of arguments, which then can be easily transformed into decision trees for classification [4]. The advantage of decision trees is their suitability for big data. Some of the mentioned disadvantages can be overcome by pruning the decision tree (see also Section 4.2).

Relational learning and inductive logic programming are concerned with the learning of first-order logic and logic program representations, respectively, which can potentially be downgraded to work on propositional logic or attribute-value representations [7]. Usually, algorithms in these fields produce monotonic rules. These do also allow for the construction of arguments, but these arguments cannot defeat each other and are

therefore less similar to everyday argumentation than arguments from defeasible rules. One possibility for simulating exceptions is to use an exception predicate for each rule that has an exception. [8] explores the theory of non-monotonic logic programming, *XHAIL* [26] and *TAL* [6] provide algorithms.

Probabilistic networks are most suitable for reasoning with uncertainty. Causal networks present an improvement over probabilistic networks (and all other methods) by taking into account the causal relationships between the variables. Causal networks also allow for counterfactual reasoning [30, ch. 13.5.2]. Moreover, experiments indicate that it is easier to reason causally than it is to reason diagnostically [17, p.121-128].

Propositionalization The representation of both data and hypotheses in Verheij's approach is restricted to propositional logic [34]. In this project we investigate an extension to input data with an attribute-value representation [7], including categorical and continuous attributes. Our approach here is to preprocess the input data by transforming continuous and categorical attributes into propositions. Some techniques for propositionalization are described in [7]. The propositionalization techniques explored in this project are Equal-Width Binning, Equal-Depth Binning, K-Means and DBSCAN, where each of the algorithms has its respective strengths and weaknesses. Equal-Width Binning and Equal-Depth Binning are the approaches with the least complexity, and K-Means and DBSCAN are more complex.

3 Preliminaries

Here, we present Verheij's approach [34], which uses the notion of a *case model* and three different notions of arguments.

Case models A case model is a description of different scenarios or situations (the cases) that can occur in the world, together with a preferences ordering over the cases denoting their relative likelihood. Each case is distinguished by the propositions that follow from it. We can alternatively define a case as the most general proposition which entails the propositions that follow from the case.

In this paper, a *case* will be a set of literals (or equivalently, a conjunction of literals). In this way, we arrive at a Boolean (propositional) representation that is suitable for machine learning. We do this by interpreting a case as a data point in the training data.

Presumption of innocence is an example of a case model from [34]. This case model has two cases, $\{innocent, \neg guilty\}$ and $\{\neg innocent, guilty, evidence\}$, where the first case is most preferred, that is, the first case has a higher probability.

Arguments An *argument* is a couple (P, C) consisting of a *premise* P and a *conclusion* C, each of which is a set of literals (or equivalently, a conjunction of literals). Note that an argument need not be valid. Verheij [34] defines the three types of arguments: *coherent* arguments, *presumptively valid* arguments and *conclusive* arguments. There holds a superset relation between the three types of arguments.

An argument is *coherent* for a case model if there exists a case in which both the premise and the conclusion are true. Note that the premise can be an empty set of literals. Examples of coherent arguments are: $(\emptyset, \{guilty\})$, $(\{evidence\}, \{\neg innocent\})$, etc.

A coherent argument is a *presumptively valid* argument if the conclusion is true in the most preferred case in which the premise is true, given the preference ordering over the cases. Note that the conclusion need not be true in less preferred cases in which the premise is true. Presumptively valid arguments are most interesting in the context of this project, since they can have exceptions and are thus very much like human arguments, which is desirable from an explainable AI perspective. We use the notation $P \rightsquigarrow C$ to denote a presumptively valid argument with premise P and conclusion C. If the premise P is an empty set of literals, the conclusion C holds by default: $\rightsquigarrow C$. Examples of presumptively arguments are: $\rightsquigarrow \{\neg guilty\}$, $\rightsquigarrow \{innocent\}$, $\{evidence\} \rightsquigarrow \{\neg innocent\}$, $\{innocent\} \rightsquigarrow \{\neg guilty\}$, etc.

An argument is *conclusive* if the conclusion is true in every case where the premise is true. Clearly, conclusive arguments are also presumptively valid. Conclusive argument need not be conclusive in the sense of everyday language because there is no formal requirement on a case model that it describes all possible cases. We use the notation $P \rightarrow C$ to denote a conclusive arguments. Examples of conclusive arguments are: $\{innocent\} \rightarrow \{\neg guilty\}$, $\{guilty\} \rightarrow \{\neg innocent\}$, etc.

4 Learning of Arguments

This section discusses the learning of arguments, specifically from data sets that specify possibly continuous values for attributes. We assume a set of attributes for which each instance of the data sets specifies the attribute values.

4.1 Discretization Techniques

With the exception of decision trees, the rule-mining algorithms in this project cannot be trained on continuous data. Therefore, in order to apply the rule-mining algorithms to data sets, we must rely on data discretization techniques to preprocess the data before mining the rules.

Equal-Width Binning This algorithm is a comparatively simple binning technique. Here, the range spanned by the smallest and largest value of a feature (referred to as *min* and *max* respectively,) is divided into a number of bins k, where each of these bins have size $\frac{max-min}{k}$. To discretize, values are assigned to the respective bin they fall into.

Equal-Depth Binning Equal-depth or equal-frequency binning is another simple discretization approach. Here, values are assigned to one of k bins, such that each bin approximately holds the same number of instances. This is done by sorting the values of the feature and assigning $\frac{n}{k}$ of the sorted instances into each bin, where n is the number of total values.

Clustering approaches To discretize more complex features in the data, clustering approaches are considered. Here, values of a given feature in the data are clustered, and replaced by the discretized value. In the data set, clusters are represented as ranges, where each cluster is described by its smallest and largest value. By the nature of the given clustering algorithms, these ranges do not overlap.

K-Means Clustering K-Means [19] is based on the idea of centroids, which are points in the centre of the cluster. Here, k centroids are initialized randomly, and the instances are assigned to the cluster whose centroid is closest. Then, the centroids are moved to the mean of the cluster, and the instances are assigned to their new cluster. The algorithm converges when the movement of centroids is below a certain threshold.

DBSCAN Clustering DBSCAN by [11] considers clusters to be regions of high density. For each instance, the algorithm counts the number of instances within a distance ϵ, also called the instance's ϵ-neighbourhood. If this number of neighbours of an instance surpasses a given threshold, the instance is considered to be a core instance, an instance within a dense region. The neighbours of this core instance are considered to be in the same cluster, where some neighbours may also be core instances themselves. Therefore, a cluster consists of a multitude of core instances.

Cluster Optimization The aforementioned clustering algorithms all provide parameters that can be tuned in order to find clusters representing the data correctly. In this project, the silhouette score introduced by [28] has been utilized to provide a metric for accuracy of clusters. This score computes the mean silhouette coefficient of all samples: $silhouette\_score = \frac{b-a}{\max(a,b)}$. Here, a denotes the mean distance to the other instances in the same cluster (intra-cluster distance) and b denotes the minimal distance to another instance that is not part of the same cluster (nearest-cluster distance).

Clusters are optimized by exhaustive search in this project, i.e., every combination of parameters is tested using the silhouette score, before returning the parameters resulting in the highest score.

4.2 Algorithms for learning arguments

We have implemented three different algorithms for learning arguments from data.[2] The first algorithm is devised by ourselves, the second one is implemented by ourselves according to the high-level description in [16], and the third one is based on the open-source library *scikit-learn* [23].

Pruned Search A naive implementation of Verheij's approach is not very efficient and has a worse case time complexity of n^k where k is the number of attributes of the data set and n is the number of bins. The *Pruned Search* algorithm improves the run-time by pruning the search space in a systematic way. This technique is known from frequent pattern mining (and its application to association rule mining [2]), and is described in the context of logical learning in [7]. The idea is to identify a quality criterion, for which the following is true: If a set fulfills the quality criterion, all its subsets must also fulfill the quality criterion. (Alternatively: If a set fulfills the quality criterion, all its *super*sets must also fulfill the quality criterion; this can be visualized by "flipping" the search space or the direction of the search). For example, in the context of frequent pattern

[2] Our code is available as an open source Python module at: https://github.com/learning-arguments/ learning_arguments

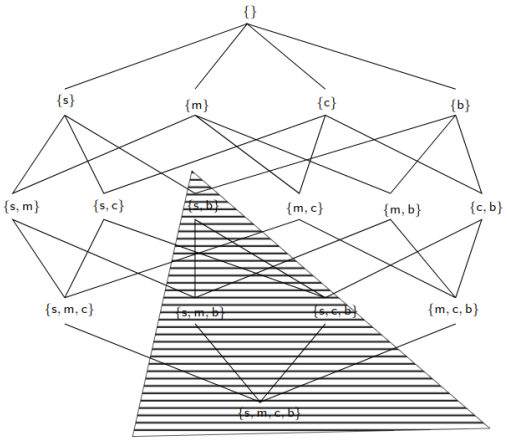


Fig. 1: *Pruning specializations.* From [7, p. 52]. All 2^n subsets of $\{s, m, c, b\}$ are systematically searched, starting from the most general set at the top. Knowing that the set $\{s, b\}$ is infrequent allows us to prune all its specializations, which is a lot.

mining, if a set is frequent, then all its subsets must also be frequent. The principle of pruning the search space is visualized in Figure 1.

This raises the issue of the selection of a suitable quality criterion for pruning the search space in our application of learning arguments. We found two quality criteria:

1. *If an argument* (P, C) *is conclusive, then all coherent arguments* (P', C) *must also be conclusive, for all* P' *that are a superset of* P*.*
2. *If an argument* (P, C) *is coherent, then all arguments* (P', C) *must also be coherent, for all* P' *that are a subset of* P*.*

The most important part of the learning algorithm in terms of efficiency is the learning of presumptively valid arguments: They are relevant for a prediction, and there are usually many more presumptively valid arguments than conclusive arguments. Unfortunately, we can prove that being presumptively valid is not a quality criterion that can be used to prune the search space. The underlying reason is that presumptively valid arguments can be overruled by more specific arguments.

Although we cannot use presumptive validity itself as a quality criterion for pruning the search space, we can at least use a condition for presumptive validity, namely: *coherence*. Our algorithm starts a search for each literal (each combination of an attribute and a bin), looking for coherent arguments with this literal as a conclusion. We search for premises with increasing number of literals. After the search is completed, we filter and merge the resulting arguments.

1. The filtering step is necessary for removing irrelevant rules. For example, when there are two arguments $a \rightsquigarrow d$ and $a \wedge b \wedge c \rightsquigarrow d$, then the second argument is more specific than the first argument and therefore only relevant if there is another

relevant argument, such as $a \wedge b \rightsquigarrow \neg d$, to which it is an exception. Generally speaking, an argument A is relevant if there is no less specific argument to A, or if A is an exception to a relevant argument. We say that an argument $P_1 \rightsquigarrow c_1$ is more specific than another argument $P_2 \rightsquigarrow c_2$, if its premise P_1 are a proper superset of the premise P_2 of the other argument.

2. During the argument generation, we only generate arguments with a conclusion of a single literal. We reduce the number of arguments for legibility by merging together any arguments (P_1, c_1), ..., (P_m, c_m) that have the same premises $P_1 = ... = P_m$ to a single argument $(P_1 = ... = P_2, \{c_1, ..., c_m\})$ with multiple conclusions.

At the end of the search step we gather all coherent arguments of the step, and check all combinations of these arguments whether their premises differ in exactly two literals. The reason is: If they are different in two literals, we can take the union of the premises as a new premise of size $i + 1$, and we know that many subsets of this premise lead to a coherent argument. Consider, for example, two premises $\{a, b, c, d\}$ and $\{b, c, d, e\}$: The union is $\{a, b, c, d, e\}$ with size $n = 5$. Enumeration shows that $2^n - 2^{n-2} = 24$ of its subsets are also subsets of at least one of the two premises of which we already know that they are coherent. It is thus much more likely for the new premise to also lead to a coherent argument than it would be for an arbitrary premise. We use this observation as a heuristic to speed up the search for other coherent arguments.

The principle of combining small sets fulfilling the quality criterion into larger sets likely to fulfill the quality criterion is known from the Apriori algorithm [32]. It makes the Apriori algorithm suitable for big data sets. Here, because coherence is only a condition but not the same as presumptive validity (which we are looking for), it at least makes the algorithm efficient enough for the medium-sized data sets we use.

An argument that is presumptively valid but not conclusive will have exceptions. We recursively search for exceptions on each presumptively valid argument, as well as exception on exceptions on exceptions etc., till a maximum specified depth.

HeRO algorithm The HeRO algorithm has been devised by [15], and the research behind it, like [34], is also originally targeted towards the legal domain [16]. It does not primarily perform a systematic search, but rather an incremental search: At each step, it considers which argument would be most valuable to be added to the theory in order to increase the accuracy the most; and then it adds the most valuable argument to the theory and asks the question again, until there is no more argument that can increase the accuracy.

The algorithm builds up a totally ordered set of arguments, and at every step it considers all positions (before, after, or between the existing arguments) for adding the next argument. For determining the most valuable argument (and its most valuable position), the criterion of *information gain*, that is increase in accuracy on the training set, is used. Similar to the Pruned Search algorithm presented above, the HeRO algorithm also starts by considering simple arguments and then in some cases also considers arguments where the premise is more specific. The mechanism for deciding whether a more specific premise should also be considered uses the criterion of *maximum information gain*. The maximum information gain of an argument is the highest information gain

that can be achieved by any argument that is more specific. This is equivalent to the information gain that would be achieved by adding an argument that correctly predicted all the rows where the premises hold.

Decision tree algorithm To classify by building tree models, the open-source library *scikit-learn* [23] is used. This implementation utilizes the CART [5] algorithm. Here, the tree is built choosing a feature k and a threshold t_k by using a cost function measuring the purity of the subsets produced by the split. In this project, this is measured by the Gini impurity introduced by [5]. Once the split has been made, the algorithm iteratively splits the subsets further, until a given maximum depth is reached, or no split reducing impurity can be found.

In a decision tree, the nodes at the bottom of the tree are referred to as leaf nodes. Trees can be converted into decision rules, where each leaf node is associated with one rule. Here, the path traversed through the decision tree represents the premise that must hold for the conclusion at the child node.

To maximize performance of the decision tree algorithm, various hyper-parameters can be tuned. In this project, this is done via Bayesian Optimization [21] utilizing the *scikit-optimize* package [1]. This algorithm samples points to construct an interpolation function, also called posterior function. This function represents the objective function (which, in this case, is a function measuring the accuracy of the tree with its parameters as inputs). New points are found using an acquisition function, which balances exploration and exploitation by calculating uncertainty in the posterior function. These query points are then used to update the posterior function. After a given number of iterations, the algorithm converges, returning an estimate of the optimal parameters by using the posterior function.

5 Experimental evaluation

5.1 Experimental setup

Legal examples We have evaluated the Pruned Search and the HeRO algorithm on legal examples described in [34] and [35].

Boston Housing Dataset We have evaluated all algorithms on the Boston Housing Dataset[3]. The Boston Housing Dataset specifies the values of 14 attributes for 506 instances. We evaluated the performance on this data set in combination with descretization algorithms. The main parameters were the number of bins used. An optimization algorithm for finding the ideal number of bins has been implemented. Because the search algorithms are very sensitive to the number of bins, we also ran the discretization algorithms with predefined number of bins, namely 2, and 4 bins.

Binning implies that several data points of the data set are grouped together. Assuming that all data points in the data set are equally likely, the number of data points that are grouped together determine the relative likelihood that we need for the case model.

[3] http://lib.stat.cmu.edu/datasets/boston

The four main steps of the experiments are data preprocessing, model training, predictions, as well as model evaluation. In a first step, the data is discretized by a method described in section 4.1. In the experiment using decision trees, only the target column is discretized. In a second step, the selected algorithm for learning the arguments from the data is applied. Afterwards, the learned model is used to generate predictions from the training data as well as the test data. Finally, the predictions are evaluated by computing accuracy and weighted F1-score. The training time is measured in order to get an understanding of the relative computational cost of the algorithms.

Parameter tuning The Pruned Search algorithm has two hyper-parameters that needed to be tuned. Next to the search depth for exception on exceptions etc., which is tested with the values 1, 5 and 20, the values 2 and 4 are tested for the maximum premise size constraint. A priori, we assume that the former will have a significant impact on the run-time while the latter will mainly determine the quality of the predictions.

Although the decision tree algorithm optimizes the parameters by Bayesian Optimization, there is still a need for specifying the parameter search space. Here, the maximum number of features randomly chosen at a split can be set between 1 and the number of features of the training data. The maximum depth is capped at 50 to retain explainability and the minimum number of samples required at a leaf node is constrained between 1 and 1000. The minimum number of samples required to split an internal node is between 2 and 1000.

The HeRO algorithm does not require any hyper-parameter tuning.

5.2 Results

Legal examples The experimental results show that the Pruned Search algorithm finds all arguments mentioned in the papers [34] and [35]. It also finds quite a few additional arguments that are correct but often irrelevant.

The HeRO algorithm generates a more concise set of arguments. However, the arguments can imply counter-intuitive self-attacks. Consider for instance the first case model in [34]: *Presumption of innocence*. This case model has two cases, $\{innocent, \neg guilty\}$ and $\{\neg innocent, guilty, evidence\}$, where the first case is most preferred. HeRO determines the following two arguments: $\rightsquigarrow innocent \wedge \neg guilty \wedge evidence$ and $evidence \rightsquigarrow \neg innocent \wedge guilty$, which imply a self-attacking argument. The first argument is counter-intuitive. HeRO determines this argument because if any information is given regarding whether or not there is evidence, then indeed it will be the information that there is evidence. Note that self-attacking arguments imply that we need to use Dung's preferred semantics [10] for determining the set of valid arguments.

Boston Housing Dataset We trained the three algorithms using 80% of the Boston Housing Dataset. The remaining 20% were used to test the models learned by the algorithms. We evaluated the algorithms on both the training and the test data set.

Decision Trees When using the Boston Housing Data set, the decision trees were scoring a perfect accuracy of 1 when using equal-depth binning or equal-width binning. Using DBSCAN gave a slightly lower accuracy of 0.99 and using k-means yielded 0.86;

Table 1: Summary of the Decision Tree results. The '–' denotes that there is no dependence on the parameter.

data type	binning method	# bins	search depth	accuracy	F1
training	kMeans	–	–	0.8613	0.8634
test	kMeans	–	–	0.8613	0.8634
training	DBSCAN	–	–	0.9975	0.9963
test	DBSCAN	–	–	0.9975	0.9963
training	other methods	–	–	1	1
test	other methods	–	–	1	1

see Table 1. Those figures showcase very well the impact and importance of choosing a good technique when discretizing the data. Note that the accuracy alone does not provide a complete picture of the quality of the algorithm: For example, using one bin for all the data would result in an accuracy of 1, yet the algorithm would not explain any structure in the data. With regard to the training time, the decision trees run significantly longer than the Pruned Search algorithms. When only discretizing the target column using equal-width binning and leaving the input values continuous, the decision trees achieve an accuracy of 0.92. This indicates that the decision trees are able to capture the structure well.

Pruned Search When it comes to Pruned Search, the results also exhibit high results for the accuracy and F1 scores. The average accuracy (F1 score) on the training set is 0.88 (0.83) and 0.85 (0.83) on the test set; see Table 2. We ran 198 experiments with the Pruned Search algorithm. The standard deviation of the evaluation metrics (accuracy: 0.0868, F1: 0.0864) indicate that the algorithms performance is rather robust.

Table 4 shows the correlation between the hyperparameters as well as the accuracy and F1 score. The results are based on the test set and training set together. When studying the correlations, we noticed that the Pruned Search algorithms do not significantly vary in accuracy and F1 score when adjusting search depth and maximum number of literals in a premise. This observation is contrary to our initial hypothesis. The positive correlation between the maximum number of literals in a premise and the run-time suggests that it increases the computational complexity. The number of bins are also positively correlated with the run-time, yet exhibit a negative correlation on the accuracy metrics. Since fewer bins make the problem easier for the algorithm, this does not come as a surprise. A very interesting observation is the negative correlation of the run-time and accuracy / F1 score. This indicates that simpler and faster algorithms perform better on this data set, likely because they are less predisposed to overfitting.

As mentioned, the discretization algorithm has a significant impact on the success of the algorithm. When looking at the situation where Pruned Search is used to mine the arguments and the number of bins is fixed to 2, one can observe that the accuracy on the test set increases when using the discretization algorithms in the following order: k-means (average accuracy 0.80), equal depth binning (0.83), equal width binning (0.91), DBSCAN (0.97). The ordering is strict, meaning that using a different discretiza-

Table 2: Summary of the Pruned Search results. The '–' denotes that there is no dependence on the parameter. The optimized number of bins is denoted 'opt' in the table. Note that the optimal number of bins may be different for each column of the data set and may depend on the binning method.

data type	binning method	# bins	search depth	max # premises	accuracy	F1
training	kMeans	2	–	–	0.7772	0.6798
test	kMeans	2	–	–	0.8039	0.7165
training	kMeans	opt	–	–	0.8515	0.8537
test	kMeans	opt	–	–	0.8168	0.8137
training	DBSCAN	–	–	–	0.9530	0.9335
test	DBSCAN	–	–	–	0.9706	0.9561
training	EWBinning	2	–	–	0.9431	0.9154
test	EWBinning	2	–	–	0.9118	0.8697
training	EWBinning	4	–	–	0.9431	0.9154
test	EWBinning	4	–	–	0.9118	0.8697
training	EWBinning	opt	–	–	0.9431	0.9154
test	EWBinning	opt	–	–	0.9118	0.8697
training	EDBinning	2	–	–	0.8317	0.8317
test	EDBinning	2	–	–	0.8333	0.8334
training	EDBinning	4	–	–	0.8243	0.8226
test	EDBinning	4	–	–	0.8725	0.8132
training	EDBinning	opt	–	–	0.8168	0.8137
test	EDBinning	opt	–	–	0.7353	0.7318

tion algorithm will always yield a higher or lower accuracy in the given settings. This emphasizes the significant impact of binning on the algorithm's performance.

HeRO The HeRO algorithm behaves similar compared to the Pruned Search algorithm in terms of performance; see Table 3. Similar as outlined above, the discretization algorithm is the main driver for the algorithm's performance. While the equal-depth binning yields an average accuracy (F1) of only 0.53 over all experiments, using k-means improves the results already significantly with an average accuracy of 0.79 (0.70). Equal-width binning further improves the situation by yielding 0.91 (0.86) and with an average accuracy 0.95 (0.93), DBSCAN gives the best results for the HeRO algorithm.

5.3 Discussion

The experiments show that arguments learned from a case model enables accurate predictions, yet needs further efforts to become practically applicable. There are two main issues that the experimental results bring to light. The first one is the exponentially increasing computational complexity of both the search and discretization algorithms. These limitations should be addressed first.

Another point worth mentioning is the binning itself. In cases where the data is binned in very few bins, it can happen that the data is heavily skewed due to outliers. When e.g. 95% of the houses are categorized as 'high price', the algorithm will score

Table 3: Summary of the HeRO results. The '–' denotes that there is no dependence on the parameter. The optimized number of bins is denoted 'opt' in the table. Note that the optimal number of bins may be different for each column of the data set and may depend on the binning method.

data type	binning method	# bins	search depth	max # premises	accuracy	F1
training	kMeans	–	–	–	0.8039	0.7165
test	kMeans	–	–	–	0.7772	0.6798
training	DBSCAN	–	–	–	0.9706	0.9561
test	DBSCAN	–	–	–	0.9455	0.9191
training	EWBinning	2	–	–	0.9431	0.9154
test	EWBinning	2	–	–	0.9118	0.8697
training	EWBinning	4	–	–	0.8861	0.8326
test	EWBinning	4	–	–	0.8725	0.8132
training	EWBinning	opt	–	–	0.9431	0.9154
test	EWBinning	opt	–	–	0.9118	0.8697
training	EDBinning	2	–	–	0.5392	0.3778
test	EDBinning	2	–	–	0.5	0.3333
training	EDBinning	4	–	–	0.5392	0.3778
test	EDBinning	4	–	–	0.5	0.3333
training	EDBinning	opt	–	–	0.5817	0.4278
test	EDBinning	opt	–	–	0.5588	0.4007

Table 4: Pruned Search Hyper-parameter Correlation Table

n=198	*Acc*	*F1*	*# bins*	*Depth*	*Run-time*	*Max # prem.*
Acc	1.000					
F1	0.940	1.000				
# bins	-0.008	0.057	1.000			
Depth	0.000	0.000	0.000	1.000		
Run-time	-0.173	-0.001	0.170	0.035	1.000	
Max # prem.	0.000	0.000			0.207	1.000

a very high accuracy with a naive prediction of always predicting 'high price'. It is obvious that the ability of the algorithms to explain patterns in data will decrease if the number of bins is reduced, while accuracy tends to increase. For that reason, just considering the accuracy might lead to false conclusions.

Furthermore, the experiments showed that simpler algorithms seem to do better than the more complex algorithms. The key takeaway from this may be that learning arguments tends to over-fit quickly.

6 Conclusion

We have implemented Verheij's approach [34] for learning arguments from a case model and showed that (1) it can reproduce the examples given in [34] and [35], and (2) it can also be used to learn arguments from a data set consisting of instances specifying

values of attributes. However, the implementation of Verheij's approach produces many correct but irrelevant arguments. A serious limitation is the run time of our implementation. To make the approach applicable to larger data sets, further research in reducing the run time is needed. Finally, the accuracy of the learned arguments for the Boston Housing Dataset depends on the used discretization algorithm with DBSCAN giving the highest performance.

We also implemented the HeRO algorithm [15] for comparison. The HeRO algorithm does not learn irrelevant arguments because it is employing the criterion of information gain. However, the learned arguments are not always intuitively plausible and may imply self attacking arguments. The accuracy of the learned arguments for the Boston Housing Dataset is 4% less compared to the implementation of Verheij's approach. Moreover, HeRO is more sensitive w.r.t. the choice of the discretization algorithm, with DBSCAN giving the best performance.

The decision tree algorithm that we implemented uses pruning on the learned tree to discard less relevant nodes. The arguments implied by the decision tree are not very intuitive. However, the decision tree algorithm reaches an accuracy of 100%.

References

1. scikit-optimize: sequential model-based optimization in python, https://scikit-optimize.github.io/stable/#
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, VLDB. vol. 1215, pp. 487–499. Citeseer (1994)
3. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82–115 (Jun 2020). https://doi.org/10.1016/j.inffus.2019.12.012
4. Breidenbach, S.: Von Text zu Code (Feb 2021)
5. Breiman, L., Friedman, J.H., Olshen, R. A.and Stone, C.J.: Classification and regression trees. (1984)
6. Corapi, D., Russo, A., Lupu, E.: Inductive logic programming as abductive search. In: Technical Communications of the 26th International Conference on Logic Programming. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2010)
7. De Raedt, L.: Logical and Relational Learning. Springer Science & Business Media (2008)
8. Dimopoulos, Y., Kakas, A.: Learning non-monotonic logic programs: Learning exceptions. In: Carbonell, J.G., Siekmann, J., Goos, G., Hartmanis, J., Leeuwen, J., Lavrac, N., Wrobel, S. (eds.) Machine Learning: ECML-95, vol. 912, pp. 122–137. Springer Berlin Heidelberg, Berlin, Heidelberg (1995). https://doi.org/10.1007/3-540-59286-5_53
9. Doshi-Velez, F., Kortz, M., Budish, R., Bavitz, C., Gershman, S., O'Brien, D., Scott, K., Schieber, S., Waldo, J., Weinberger, D., Weller, A., Wood, A.: Accountability of AI Under the Law: The Role of Explanation. arXiv:1711.01134 [cs, stat] (Dec 2019)
10. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2), 321–357 (Sep 1995). https://doi.org/10.1016/0004-3702(94)00041-X
11. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. (1996)
12. Goodman, B., Flaxman, S.: European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation". AI Magazine **38**(3), 50–57 (Oct 2017). https://doi.org/10.1609/aimag.v38i3.2741

13. Governatori, G., Stranieri, A.: Towards the application of association rules for defeasible rules discovery. Jurix 2001 pp. 63–75 (2001)
14. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques, 3rd Edition. Morgan Kaufmann (Jun 2011)
15. Johnston, B., Governatori, G.: An algorithm for the induction of defeasible logic theories from databases. In: Proceedings of the 14th Australasian Database Conference-Volume 17. pp. 75–83 (2003)
16. Johnston, B., Governatori, G.: Induction of defeasible logic theories in the legal domain. In: Proceedings of the 9th International Conference on Artificial Intelligence and Law. pp. 204–213 (2003)
17. Kahneman, D., Tversky, A.: Judgment under Uncertainty: Heuristics and Biases. Cambridge university press (1982)
18. Kakas, A., Michael, L.: Abduction and Argumentation for Explainable Machine Learning: A Position Survey. arXiv preprint arXiv:2010.12896 (2020)
19. Lloyd, S.P.: Least square quantization in pcm (1957)
20. Michael, L.: Cognitive Reasoning and Learning Mechanisms. In: AIC. pp. 2–23 (2016)
21. Mockus, J.: On bayesian methods for seeking the extremum (1974)
22. Modgil, S., Prakken, H.: The ASPIC + framework for structured argumentation: A tutorial. Argument & Computation **5**(1), 31–62 (Jan 2014). https://doi.org/10.1080/19462166.2013.869766
23. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
24. Pollock, J.L.: Defeasible reasoning. Cognitive Science **11**, 481–518 (1987)
25. Raad van State: ECLI:NL:RVS:2017:1259 (May 2017)
26. Ray, O.: Inferring process models from temporal data with abduction and induction. In: 1st Workshop on the Induction of Process Models (2007)
27. Rechtbank den Haag: ECLI:NL:RBDHA:2020:1878 (Mar 2020)
28. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987). https://doi.org/https://doi.org/10.1016/0377-0427(87)90125-7, https://www.sciencedirect.com/science/article/pii/0377042787901257
29. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River (2010)
30. Russell, S.J., Norvig, P.: Artificial Intelligence a Modern Approach. Pearson, Boston (2020)
31. Special Interest Group on Artificial Intelligence: Dutch Artificial Intelligence Manifesto. Tech. rep., ICT Research Platform Nederland, The Netherlands (Sep 2018)
32. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson, Harlow (2014)
33. Toulmin, S.: The uses of argument. Cambridge University Press (1958)
34. Verheij, B.: Proof with and without probabilities. Artificial Intelligence and Law **25**(1), 127–154 (2017)
35. Verheij, B.: Analyzing the Simonshaven case with and without probabilities. Topics in cognitive science **12**(4), 1175–1199 (2020)
36. Waltl, B., Vogl, R.: Explainable artificial intelligence the new frontier in legal informatics. Jusletter IT **4**, 1–10 (2018)
37. Witten, I.H. (ed.): Data Mining: Practical Machine Learning Tools and Techniques. Elsevier, Amsterdam (2017)

Combining Mental Models with Neural Networks

Paweł Mąka, Jelle W.M. Jansen, Theodor Antoniou, Thomas Peter Maximilian Bahne, Kevin Müller, Can Türktaş, Nico Roos, and Kurt Driessens

Data Science and Knowledge Engineering, Maastricht University, Maastricht, The Netherlands
{p.maka,jwm.jansen,t.antoniou}@student.maastrichtuniversity.nl,
{t.bahne,kevin.muller,c.turktas}@student.maastrichtuniversity.nl,
{roos,kurt.driessens}@maastrichtuniversity.nl
http://www.maastrichtuniversity.nl/dke/

Abstract. Human reasoning under uncertainty is conjectured to use Mental Models as a representation format. Each Mental Model characterizes a possible state of the world based on and constrained by the available information. Conclusion about the world must hold in each of these Mental Models. An important task in human reasoning is the construction these Mental Models using the available information. This paper investigates whether it is possible to design a neural network architecture that enables the construction of Mental Model, similarly to the conjectured way that humans reason. The paper investigates different architectures in an incremental way. The final architecture not only produces the correct mental models but also learns correct mental models for intermediate representation without being explicitly trained to do so. This contributes to the explainability of the approach.

Keywords: Machine Learning · Mental Models · Neural Networks · Reasoning

1 Introduction

Machine Learning and reasoning have been extensively researched in the past, with different attempts to combine those two research fields by encoding rule sets, with which a neural network learns the ability to reason through specified rules [4,12]. Human reasoning however, is not thought to work with a fixed set of reasoning rules that is encoded in the human natural neural network (the brain). Reasoning in the human mind comes with the concept of **Mental Models (MMs)**, a simplified representation of how humans understand the world [7]. A single MM is an abstract representation, an internal picture, of one distinct possible instantiation of the world. For example, if you are concerned if you can wear your new all-white sneakers the next day under the uncertainty of the weather, you might have two MMs for tomorrow: either it is good weather and you wear your sneakers, or it is bad weather and you will not wear your sneakers. These models allow complex situations to be simplified by getting rid of uncertainty through duplication and help make decisions based on similar

situations. Reasoning can be defined as "Algebraic manipulation of previously acquired knowledge in order to answer a new question".[1] Humans can combine two or more mental models representing pieces of information in order to reach a conclusion. For example, knowing that both "$p \vee q$" and "$\neg q$" is true, we can conclude that p must be true. We should therefore be able to generate MMs and algebraically manipulate them in a machine learning setting to produce a conclusion (possibly a single or multiple MMs). Since it is not obvious what a mathematical definition of a MM could be, the first task of this paper is to translate this concept into a machine learning setting.

This paper investigates whether it is possible to design a neural network architecture that enable the construction of Mental Model, similarly to the conjectured way that humans reason. The approach differs from, for instance, neuro-symbolic computing by not explicitly encoding knowledge in link of the neural network. All information / knowledge is provided as input to the neural network. In this investigation, we start with information formulated in Boolean algebra sentences. Of course it is not difficult to create a neural network that answers queries for such inputs. However, that is not the goal of this investigation.

The remainder of this paper is organized as follows. The next section describes related work. Section 3 defines the setting in which we do our research. Section 4 describes the neural network architectures that we have developed and Section 5 describes the experiment that we have performed with these architecture. Section 6 concludes the paper.

2 Related Work

Machine Learning and reasoning originated as separate research fields of Artificial Intelligence in the past, but have recently seen different approaches of combining those in conjuncted research [4,12]. The research field of Neural-Symbolic Computing aims to embed the two most fundamental human cognitive abilities into a system: "the ability to learn from the environment, and the ability to reason from what has been learned." [4] They make use of neural networks, by encoding reasoning rules in between the layers of a neural network.

However, it has been argued that humans reason with the use of MMs, aiming to find conclusions that are true [9]. Those conclusions can be an outcome of a conjunction or a repetition of the premise concerned. Humans search for relations that are not explicitly asserted in the premises, reaching conclusions that seem the most probable [8]. Since the implementation of MMs in this project does not hard-code reasoning rules in between the layers of a neural network, it adds to current research by investigating a more general and flexible approach to reasoning in neural networks.

Since MMs are an integral component of this paper, we repeat some of the theory on MMs. As Johnson-Laird explains, "[...], each mental model represents what is common to a distinct set of possibilities." [8, p. 2]. They do not reflect every detail of that distinct state of the world, but reduce the available information to the aspects necessary for the context. In the sneaker example that

was given in Section 1, the only necessary information in the context of deciding whether or not to wear new, white sneakers (without getting them dirty on the first day) are the weather conditions, which can be good or bad in this case. All other information that might be available is not reflected in the two MMs that arise from this context. It does not matter if the person had one or two cups of coffee in the morning, it is irrelevant what was in TV the night before. MMs reduce the mental load by excluding unnecessary information.

To further reduce load on our working memory, humans build MMs on the principle of "truth" [9]. The principle of truth dictates that MMs only represent propositions of the premise that are true and neglect those that are false, i.e. they follow the closed world assumption. For instance, when considering the exclusive disjunction "I can go on holiday or else I can finish the project I am working on", humans would build two MMs according to the principle of truth: "I go on holiday" and "I finish my project". Observe that each model does not include the falsification of the respective other premise. If one would to be precise and construct complete models, we would get "I go on holiday and I don't finish my project" and "I do not go on holiday and I finish my project". However, this shortcut used by our brains results in predictable misjudgement in deduction, which we do not want to imitate with neural networks. Therefore, we will disregard the principle of truth when constructing the MMs for our networks in Section 3.

Another assumption of MM theory states that MMs are iconic. "The structure of a [Mental Model] representation corresponds to the struc-ture of what it represents."[8, p. 2]. This is intuitive, as we think about different topics in different ways. The concept of biological evolution for example is entirely different from theory on radioactive decay. The considerations and dependencies that need to be taken into account greatly change from one context to another, implying that the structure of corresponding MMs also differ. This paper will take all of the three mentioned aspects of MM theory into account when defining the MMs in the next section.

3 Defining the Setting

In this section, the representation of MMs used in this paper will be described. As stipulated by iconicity, a MM needs to resemble the structure of what it represents. Therefore, it is necessary to first fix the context, i.e. whatever it is that the MM should represent. For this paper, a MM will be defined in the context of boolean algebra, because of its scalability, modularity [1], and relative simplicity. The sentences used in the datasets are composed of two simple sub-sentences, which are both assumed to be true. To increase complexity, sub-sentences can be combined with the "and" operator in order to obtain a single more complex sentence. This process can be repeated for further complexity.

[1] This results from the fact that any sentence can be transformed into conjunctive normal form [11, p. 253]

The truth table of a logical sentence represents a possible state of the world and can be interpreted as a MM, which allows us to think about the information contained in a logical sentence and helps us to perform further reasoning tasks. Even though the human mind might not need to fall back on truth tables and rather represents Boolean algebra in a more advanced way [9, p. 114], this does not compromise the validity of using truth tables for this context.

There is one modification to traditional truth tables used in this paper: when a variable does not appear in a sentence, or if the value of the variable does not influence the value of the sentence given the other variables, a value of "none" will be assigned to this variable, instead of "true" or "false". This modification is again inspired by MM theory, which states that MMs reduce the amount of stored information to a minimum in order to preserve cognitive capacity. The value of a variable that does not appear in a sentence has no effect on the value of the sentence. Hence, a single MM which assigns such variable the "none" value contains the same information of two other MMs which are identical to the first, except that the value of the variable is now changed to "true" and "false" respectively.

4 Methodology

This section describes the datasets we created to evaluate the architectures on a Boolean algebra reasoning task and gives a detailed description of the created neural networks. The neural networks can be divided into architectures predicting conclusion in the form of one MM or multiple MMs. This is also reflected in the structure of the datasets.

4.1 Creating the Learning Data

The datasets consist of inputs in the form of *two* logical sub-sentences[2] (Boolean expressions) and labels that represent a single or multiple MMs induced by both of the sub-sentences being true. For all datasets in this paper, it is always assumed that the logical sentence (or both sub-sentences) given as input is "true".

When creating the datasets a parameter representing the "depth" of a logical sentence is used. A sentence can be represented as a tree-structure with the variables in the leafs and non-terminal nodes containing logical operators. This means that a sentence of depth 1 consists of at most one logical operator and two variables. Examples for sentences of depth 1 are "x_1 or x_2" and "not x_3"; and of depth 2 are "x_1 or not x_2" and "(x_1 or x_2) and x_3".

The labels (MMs) use a vector representation. The length of the vector n corresponds to the number of logical variables used in the dataset. In the experiments $n = 5$, the logical variables are denoted with symbols x_1-x_5. Each element in the vector encodes the value of one variable in the MM. A value of 1 indicates

[2] If the evaluated neural network requires exactly one sentence as the input, the two sub-sentences can be concatenated with the "and" operator between them.

that the variable is true in the MM, the value -1 indicates false. Additionally, a variable in the vector can be assigned a value of 0, which corresponds to the "none" value described in Section 3.

All datasets were created by implementing the generate-and-test approach. We randomly generate a sentence and algorithmically determine what MMs are compatible with the sentence. If both the sentence and the resulting MMs (conclusion) fulfil the specifications, the sentence is added to the dataset. These steps were performed a specified number of times.

The first two datasets are called **Many-to-Single MM Small** and **Many-to-Single MM Big** respectively. Examples in these datasets induce one single MM, where any number of variables can be true or false (as long as at least one variable is not "none"). Additionally, The difference between the small and big version is the depth of the two sub-sentences. In Many-to-Single MM Small dataset, sub-sentences have a maximum depth of 1. This leads to a combined sentence of a maximum length of 11 (the length of the sequence of variables, operators and brackets that forms a sentence) with the sub-sentences of length 5 at most. This dataset contains 2369 sentences (consisting of two sub-sentences each). On the other hand, sub-sentences in the Many-to-Single MM Big dataset can have a maximum depth of 2 therefore the maximum length of a sentence is increased to 27. Each subsentence has $2 \cdot 13$ (including the outer brackets) plus 1 for the combining "and" operator. The size of this dataset is 276178 sentences and labels. The third dataset is called **Many-to-Many MM**. For this dataset, no restrictions were put on number of the induced MMs. Again, sentences are made up of two conjuncted sub-sentences, each of depth 2 at maximum. This dataset contains 3489 datapoints. Both Many-to-Single MM Small and Many-to-Many MM datasets consist of all possible sentences fulfilling the conditions (depth and number of conclusion MMs). The size of Many-to-Single MM Big dataset results from running the generate-and-test algorithm until less than 1% of generated sentences were not already included in the dataset.

4.2 Constructing the Architecture

The goal of our design is to encourage the network to not only output vectors which we interpret as MMs, but also to internally use these MMs in order to derive the desired output. In addition to testing if we can increase performance this way, we hope to achieve greater interpretability for the internal reasoning process of the neural network.

For the task of predicting MMs we designed a neural network that accepts two logical sub-sentences as input and predict a single or multiple mental models that are a conclusion of the input (with the assumption that both the sub-sentences are true). The first network we implement in this context is called **Single-mental model Net (Single-mmNet)** and is trained on the Many-to-single MM datasets. In addition, we define two other networks, that go by the names of **Multi-mental model Net with direct input (Multi-mmNet direct)**, and **Multi-mental model Net combination (Multi-mmNet combination)**. These networks are trained on the Many-to-multiple MM dataset.

To encourage our networks to internally use MMs, sub-sentences are fed into a shared sub-sentence encoder, before a final reasoning module combines the outputs for each sub-sequence (see Figure 1). Conceptually, we want our network to generate the MMs for every sub-sentence before merging the information in those sub-sentences in the final reasoning module. This concept is rooted in the modularity property described in Section 3. We introduce the inference layer that combines the MMs induced by sub-sentences. While not explicitly forced into a specific representation for sub-sentence MMs, the sub-sentence encoder adopts our definition of MMs during training.

To implement this architecture, it is sufficient to use a simple feed-forward architecture with an embedding layer and one fully-connected hidden layer. The activation functions were set to hyperbolic tangent for the output layer and ReLU for the hidden layer. The output is reshaped to a matrix $Y \in \mathbb{R}^{M \times D}$, where M is a constant number of MMs (specified as a hyper-parameter) and D is the number of logic variables (five in our case).

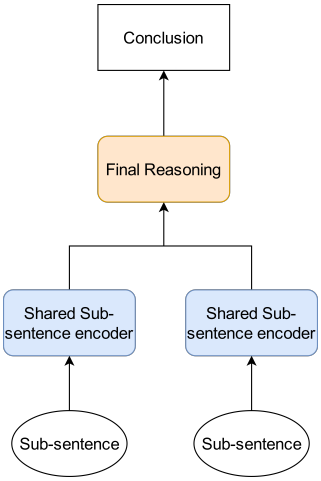


Fig. 1: General architecture of neural network consisting of shared sub-sentence encoders and a final reasoning module

Predicting a Single Mental Model The Single-mmNet architecture uses a fully connected network as a sub-sentence encoder and the reasoning module described in detail below. It is trained end-to-end using standard gradient-based optimization on the Many-to-single MM dataset.

The reasoning module dubbed "MM-Inference Layer" is meant to combine two outputs of the sub-sentence encoder (one output for every sub-sentence). The inference layer accepts two matrices, each representing the MMs induced by one sub-sentence. We denote the matrices with $Y^1 \in \mathbb{R}^{M \times D}$ and $Y^2 \in \mathbb{R}^{N \times D}$, where

M and N are the numbers of MMs induced by the first and second sub-sequence respectively, and D is the number of variables. It is assumed that all elements of the two matrices are in the range $[-1, 1]$. Let Y^1_m for $m \in \{1, 2, \ldots, M\}$ denote a MM model from the first sub-sentence (i.e. a row of Y^1) and Y^2_n for $n \in \{1, 2, \ldots, N\}$ a MM model from the second sub-sentence (i.e. a row of Y^2). For every pairing $m,\ n$, we calculate two quantities: $V_{m,n} \in \mathbb{R}^D$ and $C_{m,n} \in \mathbb{R}$, which we call *value* and *correctness*. To obtain the *value* $V_{m,n}$ between two MMs, we simply add the two MMs element wise and "clamp"[3] the resulting numbers between -1 and 1.

This approach disregards the situation when two MMs are incompatible with each other. Two MMs are incompatible when the same variable is true in one model and false in the other. (We will sometimes refer to such a variable as an incompatible variable). To indicate when two MMs are incompatible, we introduce *correctness*. For "perfect" values for variables of either exactly -1, 0, or 1, the *correctness* of a pair of MMs will be 1 if the two models are compatible (i.e. no variable is true in one of the models and false in the other) and 0 otherwise. During training and testing however, the sub-sentence encoder could assign any value between -1 and 1 to the variables. As a consequence, the correctness becomes a number between 0 and 1. Therefore in practice, two MMs become increasingly incompatible, as the absolute difference between the variables increases. The two quantities (value and correctness) are calculated using Eq. 1 and 2 respectively.

$$\begin{aligned} &V_{m,n} = \min(1,\ \max(-1, Y^1_m + Y^2_n)) \\ &\forall m = 1, ..., M, \quad \forall n = 1, ..., N \end{aligned} \tag{1}$$

$$\begin{aligned} &C_{m,n} = \prod_{d=1}^{D} [1 - \max(0,\ |Y^1_m - Y^2_n| - 1)]_d \\ &\forall m = 1, ..., M, n = 1, ..., N \end{aligned} \tag{2}$$

The Single-mmNet network only outputs one MM, which is calculated using Eq. 3. In essence, Z is a sum of all values weighted with their respective correctness and normalised by the sum of all correctness.

$$Z = \frac{\sum_{m=1,n=1}^{M,N} V_{m,n} \cdot C_{m,n}}{\sum_{m=1,n=1}^{M,N} C_{m,n}} \tag{3}$$

The use of a fully-connected network as a sub-sentence encoder means that the number of MMs is set beforehand and identical for each sub-sentence. To allow a variable number of sub-sentence MMs, we added a second type of output to the fully-connected network - scores $S^1 \in \mathbb{R}^M$ and $S^2 \in \mathbb{R}^N$ for the first and second sub-sentence respectively. Each MM has a corresponding score, where the value of 1 indicates that the MM is correct and contains important information and value of 0 means that the MM is erroneous or redundant. In contrast to value

[3] Clamping indicates setting all values < -1 to -1 and all values > 1 to 1

and correctness, scores are taken directly from the outputs of sub-sentence encoders. The MM-Inference Layer is subsequently adapted to accept these scores as additional inputs and take them into account when calculating the output - Eq. 4 shows the formulation used. The introduction of scores does not change the dimension of the output $Z^s \in \mathbb{R}^D$. It was empirically found that normalizing the sum by the summed correctness (hence without scores) yields more stable results in terms of test accuracy.

$$Z^s = \frac{\sum_{m=1,n=1}^{M,N} V_{m,n} \cdot C_{m,n} \cdot S_m^1 \cdot S_n^2}{\sum_{m=1,n=1}^{M,N} C_{m,n}} \tag{4}$$

Predicting Multiple Mental Models Allowing more MMs as a conclusion is inherently a many-to-many problem. For this problem we propose two encoder-decoder architectures built around a modified version of the MM-Inference Layer for the encoder part, and an LSTM layer for the decoder part. Both models use the same shared fully-connected sub-sentence encoder with scores. The MM-Inference Layer is modified to produce values (see Eq. 1) and scores based on correctness and input scores as defined in Eq. 5.

$$S_{m,n} = C_{m,n} \cdot S_m^1 \cdot S_n^2 \tag{5}$$

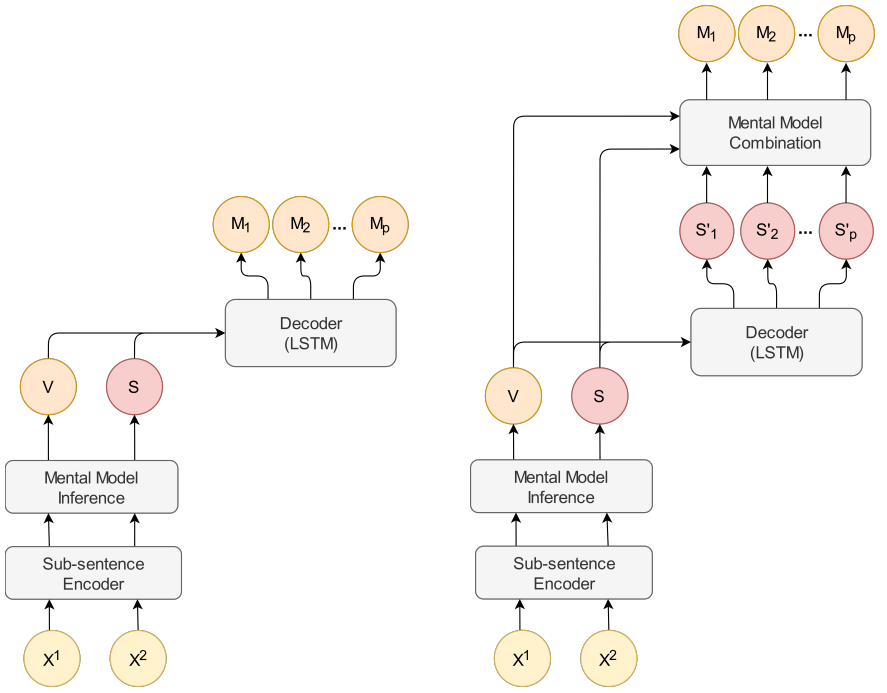
The output values V and scores S are flattened and concatenated, and are used as the initial hidden state and cell state of the LSTM in the decoder part. The output of the LSTM feeds into the fully-connected layer.

In the *first* architecture **Multi-mmNet (direct output)** the fully-connected layer has a number of neurons equal to the number of variables n and uses a hyperbolic tangent activation function. The outputs of this layer are interpreted as predicted MMs (see Figure 2a), and are auto-regressively fed back as the input for the prediction of the next MM. We stop the model when the end-of-sequence token is reached.

In the *second* architecture **Multi-mmNet (combination)** the fully-connected layer has a number of neurons equal to the number of mental models returned by the encoder ($M \cdot N$), and a sigmoid activation function. Its output S' is interpreted as scores for combining the MMs obtained from the MM-Inference Layer of the decoder part. This combination happens through a MM-Combination Layer that computes the sum of MMs weighted by the scores predicted by the decoder (see Figure 2b). The following shows the calculation for the p-th output of the network:

$$Z_p^c = \frac{\sum_{i=1}^{M \cdot N} V_i \cdot S_i \cdot S'_{p,i}}{\max(1, \sum_{j=1}^{M \cdot N} S_j \cdot S'_{p,j})}. \tag{6}$$

To avoid division by 0, we do not allow the deliminator to be less than 1. During training we use teacher forcing [14,5], where the training data is used as the input to the decoder instead of the output generated by the network in the previous step. During inference the resulting output is used auto-regressively



(a) Multi-mmNet (direct output) (b) Multi-mmNet (combination)

Fig. 2: Schematic connections of Multi-mmNet (direct output) architecture on the left and Multi-mmNet (combination) on the right, where X^1 and X^2 are the input sub-sentences, M_1 to M_p are the output MMs, and S'_1 to S'_p are the scores returned by the decoder of the Multi-mmNet (combination).

to predict the next output, until the end-of-sequence token is reached. This technique is used to mitigate the network instability, and make it converge faster. In the experiments, we chose a MM containing only 0s as the end-of-sequence token for both architectures.

5 Experiments

Each dataset is split into training, validation and test subsets according to 80%, 10% and 10% ratios respectively. Training used the Adam optimizer [10] and a mean squared error as loss function for all models. Training was terminated early based on the validation loss.

5.1 Many-to-single Mental Model Architectures

As discussed above the Single-mmNet architecture predicts one MM given two sub-sentences. The number of mental models of the fully-connected sub-sentence

encoder was set to 3 for the Many-to-Single MM Small dataset experiments. For the Many-to-Single MM Big dataset it was experimentally found that using 6 mental models resulted in the best performance. The network was evaluated both without and with the use of scores. With the use of scores, the Single-mmNet with scores theoretically has the ability to distinguish which MM is important and which is not. In practice however, scores converge to 1.0 and are not used by the network as intended. For comparison we trained a standard LSTM [6] on the datasets concatenating the two sub-sentences using an *and* operator. We choose the LSTM as baseline after empirically comparing alternatives on the simple task of predicting a single true variable in a logical sentence.[4]

Results reported on the small version Many-to-Single MM dataset summarise 6 repetitions of the experiment, while those on the big version stem from 3 repetitions. Because Single-mmNet with scores performed better than the one without, only this architecture was evaluated on the big dataset.

Observations and Discussion Table 1 summarises the performances. The addition of scores seems to increase the robustness of the single-mmNet architecture, despite the fact that they converged to 1.0 during training. The model with scores achieved similar accuracy as the LSTM.

Table 1: Experiments with the small and big version of the Many-to-single MM dataset

Model	**Average Accuracy**	
	Small Dataset	**Big Dataset**
LSTM	100%	99.80%
Single-mmNet without scores	96.84%	-
Single-mmNet with scores	100%	99.96%

The LSTM and Single-mmNet with scores both reached perfect accuracy on the small version of the dataset. They also performed very well on the big dataset with an average accuracy of 99.80% and 99.96% respectively. The Single-mmNet without scores reached this perfect accuracy only in some runs (on the small dataset).

While the scores don't seem to fulfill the purpose of indicating which MM is relevant, their use improved accuracy and stability of the architecture, possibly

[4] The LSTM outperformed a Vanilla RNN [3], a GRU [2], and a simplified Transformer [13] that consisted of the encoder part of the Transformer with a fully-connected output layer.

being of use during the training phase, so it was decided to keep them in the subsequent, more complex architectures as well.

5.2 Many-to-many Mental Model Architectures

Many sentences imply multiple MMs, so the Many-to-Many MM dataset was used to reflect this fact. Based on the performance of the Vanilla LSTM, the Single-mmNet architecture was expanded by feeding the encodings into an LSTM decoder as discussed in Section 4.2. The MMs and scores are used as initialisation of the LSTM-decoder, which outputs MMs directly for "Multi-mmNet (direct output)" or outputs scores used in the Mental Model Combination Layer for "Multi-mmNet (combination)". The number of mental models of the fully-connected sub-sentence encoder was once again set to 3.

As benchmarks, we used encoder-decoder networks based on an LSTM. The first two architectures use sub-sentence representations where one uses specific Start of Sequence (SOS) and End of Sequence (EOS) tags, while the other does not. The third model uses a symbolic concatenation of the two sub-sentences and without specific SOS/EOS tags.

Observations and Discussion The performance of the Multi-mmNet architectures (both "direct output" and "combination") and benchmark networks can been seen in Table 2. These results are achieved after fine-tuning the models' parameters. A perfect accuracy was achieved by the model outputting MMs and an average 99.67% accuracy by the model outputting scores for combining MMs, actually reaching perfect accuracy 4 out of 6 times.

The two LSTM networks using sub-sentences exhibit the similar accuracy, while the encoder-decoder model using symbols performed even slightly better, but the differences are very small. In fact, all models performed comparable to the benchmarks when judging accuracy. Beside predicting the correct MM, the model predicts the MMs in the same order they were listed in the dataset. This is an expected result for an LSTM network.

Table 2: Experiments with Many to Many dataset

Model	**Average Accuracy**
LSTM – Encoder decoder sub sentences no start index	99.70%
LSTM – Encoder decoder sub sentences with start index	99.70%
LSTM – Encoder decoder symbol no index	99.95%
Multi-mmNet (direct output)	100%
Multi-mmNet (combination)	99.67%

Contrary to single-mmNet with scores however (see Section 5.1) Multi-mmNet (combination) *does* use the scores to mark the importance of the MMs. This was not observed for the Multi-mmNet (direct output) - the sub-sentence encoder of this model did not output MMs at all. Using MMs as output of the decoder, resulted in reverting back to an uninterpretable latent representation of sub-sentences, not unlike the LSTM benchmark models. In case of Multi-mmNet (combination) however, without being explicitly trained to do so, the sub-sentence encoder produced MMs as we hypothesized they could be used.

5.3 Sub-sentence encodings

To illustrate the MMs (and scores if applicable) produced by the fully-connected sub-sentence encoder, Table 3 shows the rounded outputs for a selection of sub-sentences. The outputs of Single-mmNet without scores are easily interpreted as MMs corresponding to the sub-sentence. When only one MM is sufficient to represent the information in the sub-sentence, it is copied to all three outputs. Despite using scores to allow the network to use less MMs for each sub-sentence in the Single-mmNet with scores, the network learned to output all scores as 1.

The encoder of Multi-mmNet (direct output) did not learn to output MMs at all. Although the network achieves perfect accuracy, the output of the sub-sentence encoder is not easily interpreted: the vectors do not correspond to MMs of sub-sentences, with scores close to 0 for all outputs. The introduction of the Mental Model Combination layer in Multi-mmNet (combination) enabled sub-sentence encoder to output MMs, and subsequently improved the interpretability of the encodings. Additionally, the encoder learned to use less outputs by setting corresponding scores to 0. That said, the encoder still sets two scores to 1 for most of the sub-sentences, therefore the duplication of MMs is still present in the output.

The networks were trained in end-to-end fashion and were not directly optimized to internally employ MMs. The usage of MMs as an intermediate representation is imposed through MM-Inference Layer in all three architectures exhibiting this behaviour. In case of these architectures - and in contrast to the Multi-mmNet (direct output) - this layer is the last (output) layer of the networks, which were trained to predict MMs. The layer preserves the diminsionality of the input as it is being processed, and the processing itself was designed utilize of the semantics of the introduced representation of MMs. This leads to a substantial improvement for the interpretability of the latent space of the proposed architectures.

6 Conclusion

This paper investigated enabling neural networks to make use of Mental Models for solving reasoning tasks. We conclude that it is possible to construct and train neural network architecture to generate Mental Models for the input information. This can been done by introducing vector encoding of Mental Models,

Table 3: Rounded output of the sub-sentence encoder in MM architectures

Architecture	Sub-sentence	Y_1	S_1	Y_2	S_2	Y_3	S_3
Single-mmNet without scores	(x_2 or x_1)	[1, 1, 0, 0, 0]	-	[-1, 1, 0, 0, 0]	-	[1, -1, 0, 0, 0]	-
	not x_1	[-1, 0, 0, 0, 0]	-	[-1, 0, 0, 0, 0]	-	[-1, 0, 0, 0, 0]	-
	x_5	[0, 0, 0, 0, 1]	-	[0, 0, 0, 0, 1]	-	[0, 0, 0, 0, 1]	-
	(x_1 and x_5)	[1, 0, 0, 0, 1]	-	[1, 0, 0, 0, 1]	-	[1, 0, 0, 0, 1]	-
	(x_3 and x_2)	[0, 1, 1, 0, 0]	-	[0, 1, 1, 0, 0]	-	[0, 1, 1, 0, 0]	-
	(x_2 or x_1)	[1, 1, 0, 0, 0]	-	[-1, 1, 0, 0, 0]	-	[1, -1, 0, 0, 0]	-
	(x_1 and x_3)	[1, 0, 1, 0, 0]	-	[1, 0, 1, 0, 0]	-	[1, 0, 1, 0, 0]	-
	not x_3	[0, 0, -1, 0, 0]	-	[0, 0, -1, 0, 0]	-	[0, 0, -1, 0, 0]	-
	x_1	[1, 0, 0, 0, 0]	-	[1, 0, 0, 0, 0]	-	[1, 0, 0, 0, 0]	-
Single-mmNet with scores	(x_2 or x_1)	[1, -1, 0, 0, 0]	1	[-1, 1, 0, 0, 0]	1	[1, 1, 0, 0, 0]	1
	not x_1	[-1, 0, 0, 0, 0]	1	[-1, 0, 0, 0, 0]	1	[-1, 0, 0, 0, 0]	1
	x_5	[0, 0, 0, 0, 1]	1	[0, 0, 0, 0, 1]	1	[0, 0, 0, 0, 1]	1
	(x_1 and x_5)	[1, 0, 0, 0, 1]	1	[1, 0, 0, 0, 1]	1	[1, 0, 0, 0, 1]	1
	(x_3 and x_2)	[0, 1, 1, 0, 0]	1	[0, 1, 1, 0, 0]	1	[0, 1, 1, 0, 0]	1
	(x_2 or x_1)	[1, -1, 0, 0, 0]	1	[-1, 1, 0, 0, 0]	1	[1, 1, 0, 0, 0]	1
	(x_1 and x_3)	[1, 0, 1, 0, 0]	1	[1, 0, 1, 0, 0]	1	[1, 0, 1, 0, 0]	1
	not x_3	[0, 0, -1, 0, 0]	1	[0, 0, -1, 0, 0]	1	[0, 0, -1, 0, 0]	1
	x_1	[1, 0, 0, 0, 0]	1	[1, 0, 0, 0, 0]	1	[1, 0, 0, 0, 0]	1
Multi-mmNet (direct output)	(x_2 or x_1)	[1, 0, 0, -1, -1]	0	[0, 1, 0, -1, 0]	0	[0, -1, 1, 0, 0]	0
	not x_1	[1, 1, -1, 1, -1]	0	[-1, 1, 1, 0, 0]	1	[-1, 1, 0, 1, 1]	0
	x_5	[1, 0, 1, 1, -1]	1	[1, 0, 0, 1, 0]	1	[0, 1, 1, 1, 0]	0
	(x_1 and x_5)	[0, -1, 1, 1, -1]	1	[1, 0, 0, 1, 1]	1	[1, -1, 1, 1, 0]	1
	(x_3 and x_2)	[1, 1, -1, -1, -1]	0	[1, 1, -1, 0, 0]	0	[1, 1, -1, 0, -1]	1
	(x_2 or x_1)	[1, 0, 0, -1, -1]	0	[0, 1, 0, -1, 0]	0	[0, -1, 1, 0, 0]	0
	(x_1 and x_3)	[-1, 1, -1, 0, -1]	1	[1, 0, 0, 1, 1]	1	[1, -1, -1, 0, -1]	1
	not x_3	[0, -1, 1, -1, -1]	1	[-1, 0, 1, 0, 0]	1	[0, 1, 1, 0, 1]	0
	x_1	[-1, 0, 0, -1, -1]	1	[1, 0, 0, 1, 1]	1	[1, -1, 1, 0, 0]	1
Multi-mmNet (combination)	(x_2 or x_1)	[1, 0, 0, 0, 0]	1	[-1, 1, 0, 0, 0]	1	[1, 1, 0, 0, 0]	0
	not x_1	[0, 1, 1, 0, 1]	0	[-1, 0, 0, 0, 0]	1	[-1, 0, 0, 0, 0]	1
	x_5	[-1, 1, 1, 1, 0]	0	[0, 0, 0, 0, 1]	1	[0, 0, 0, 0, 1]	1
	(x_1 and x_5)	[1, 1, 1, 1, 1]	0	[1, 0, 0, 0, 1]	1	[1, 0, 0, 0, 1]	1
	(x_3 and x_2)	[-1, 1, 1, 1, 1]	0	[0, 1, 1, 0, 0]	1	[0, 1, 1, 0, 0]	1
	(x_2 or x_1)	[1, 0, 0, 0, 0]	1	[-1, 1, 0, 0, 0]	1	[1, 1, 0, 0, 0]	0
	(x_1 and x_3)	[1, 1, 1, 1, 1]	0	[1, 0, 1, 0, 0]	1	[1, 0, 1, 0, 0]	1
	not x_3	[0, 1, 1, 0, 0]	0	[0, 0, -1, 0, 0]	1	[0, 0, -1, 0, 0]	1
	x_1	[1, 1, 1, 0, 1]	0	[1, 0, 0, 0, 0]	1	[1, 0, 0, 0, 0]	1

and formulating neural network layers that perform differentiable operations to combine those encodings. By incorporating those layers with existing neural networks, we created several architectures and trained them using gradient-based methods for the Boolean algebra reasoning tasks. All proposed neural networks achieved accuracy comparable to existing architectures. Additionally, three out of four architectures exhibited the internal usage of Mental Models in the latent space (the exception was Multi-mmNet with direct output). Only when there exists a direct path through the MM layers to the output, which is also an en-

coded MM, we observe that the simple sub-sentence encoder learned to output human-interpretable encodings - even though we trained the architectures in the end-to-end fashion. We see this fact as the advantage of those architectures as it can be used to achieve greater explainability of neural networks. The code-base of the project can be found on Github. [5]

Currently mental models are being processed in a specific order in the neural networks. The networks are good at predicting what to expect. However, in real world problems, the order of the mental models is irrelevant. This could be solved by using a permutation invariant loss function but is left for future work. A restriction of our research is how this theoretical setting can be translated to a real world problem. In this work, a specific Boolean algebra problem was explored. The presented experiments were intended as a proof-of-concept and the experiments involving larger datasets (in terms of both the number of variables and the depth of the Boolean expressions) should be conducted. The difficulties could arise when the architectures are adapted to accept other forms of input (ultimately, natural language). Additionally, our architecture is limited to reason from exactly two sub-sentences. This is left for future research.

References

1. Bottou, L.: From machine learning to machine reasoning. Machine Learning **94**(2), 133–149 (Feb 2014). https://doi.org/10.1007/s10994-013-5335-x
2. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1724–1734. Association for Computational Linguistics (Oct 2014). https://doi.org/10.3115/v1/D14-1179
3. Elman, J.L.: Finding structure in time. Cognitive Science **14**(2), 179–211 (1990). https://doi.org/10.1016/0364-0213(90)90002-E
4. Garcez, A., Gori, M., Lamb, L., Serafini, L., Spranger, M., Tran, S.: Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. Journal of Applied Logics – IfCoLog Journal of Logic and their Applications (FLAP) **6**, 611–632 (2019)
5. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), `http://www.deeplearningbook.org`
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**, 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
7. Johnson-Laird, P.: The history of mental models, pp. 179–212 (09 2004). https://doi.org/10.4324/9780203506936
8. Johnson-Laird, P.N.: Mental models and human reasoning. Proceedings of the National Academy of Sciences **107**(43), 18243–18250 (2010). https://doi.org/10.1073/pnas.1012933107
9. Johnson-Laird, P.N.: How we reason. Oxford University Press, USA (2006). https://doi.org/10.1093/acprof:oso/9780199551330.003.0028

[5] `github.com/Pawel-M/Machine-Learning-and-Reasoning`

10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR (2015)
11. Russell, S., Norvig, P.: Artificial intelligence: a modern approach (2002). https://doi.org/10.1145/201977.201989
12. Shi, S., Chen, H., Ma, W., Mao, J., Zhang, M., Zhang, Y.: Neural logic reasoning. p. 1365–1374. CIKM '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3340531.3411949
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
14. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Computation **1**(2), 270–280 (1989). https://doi.org/10.1162/neco.1989.1.2.270

SHACL: A Description Logic in Disguise

Bart Bogaerts[1], Maxime Jakubowski[2,1], and Jan Van den Bussche[2]

[1] Vrije Universiteit Brussel, Belgium
{bart.bogaerts,maxime.jakubowski}@vub.be
[2] Universiteit Hasselt, Hasselt, Belgium
{maxime.jakubowski,jan.vandenbussche}@uhasselt.be

Abstract. SHACL is a W3C-proposed language for expressing structural constraints on RDF graphs. In recent years, SHACL's popularity has risen quickly. This rise in popularity comes with questions related to its place in the semantic web, particularly about its relation to OWL (the de facto standard for expressing ontological information on the web) and description logics (which form the formal foundations of OWL). We answer these questions by arguing that *SHACL is in fact a description logic*. On the one hand, our answer is surprisingly simple, some might even say obvious. But, on the other hand, our answer is also controversial. By resolving this issue once and for all, we establish the field of description logics as the solid formal foundations of SHACL.

Keywords: Shapes · SHACL · Description Logics · Ontologies.

1 Introduction

The Resource Description Framework (RDF [17]) is a standard format for publishing data on the web. RDF represents information in the form of directed graphs, where labeled edges indicate properties of nodes. To facilitate more effective access and exchange, it is important for a consumer of an RDF graph to know what properties to expect, or, more generally, to be able to rely on certain structural constraints that the graph is guaranteed to satisfy. We therefore need a declarative language in which such constraints can be expressed formally.

Two prominent proposals in this vein have been ShEx [5] and SHACL [19]. In both approaches, a formula expressing the presence (or absence) of certain properties of a node (or its neighbors) is referred to as a "shape". In this paper, we adopt the elegant formalization of shapes in SHACL proposed by Corman, Reutter and Savkovic [6]. That work has revealed a striking similarity between *shapes* and *concept expressions*, familiar from description logics (DLs) [4].

The similarity between SHACL and DLs runs even deeper when we account for *named shapes* and *targeting*, which is the actual mechanism to express constraints on an RDF graph using shapes. A *shape schema* is essentially a finite list of shapes, where each shape ϕ_s is given a name s and additionally associated with a target query q_s. The shape–name combinations in a shape schema specify, in DL terminology, an *acyclyc TBox* consisting of all the formulas

$$s \equiv \phi_s.$$

Given an RDF graph G, this acyclic TBox determines a unique interpretation of sets of nodes to shape names s. We then say that G *conforms* to the schema if for each query q_s, each node v returned by q_s on G satisfies s in the extension of G.

Now interestingly, the types of target queries q considered for this purpose in SHACL as well as in ShEx, actually correspond to simple cases of shapes ϕ_{q_s} and the actual integrity constraint thus becomes

$$\phi_{q_s} \sqsubseteq s.$$

As such, in description logic terminology, a shape schema consists of two parts: an acyclic TBox (defining the shapes in terms of the given input graph) and a general TBox (containing the actual integrity constraints).

2 The Wedge

Despite the strong similarity between SHACL and DLs, and despite the fact that in a couple of papers, SHACL has been formalized in a way that is extremely similar to description logics [6,2,11], this connection is not recognized in the community. In fact, some important stakeholders in SHACL recently even wrote the following in a blog post explaining why they use SHACL, rather than OWL:

> OWL was inspired by and designed to exploit 20+ years of research in Description Logics (DL). This is a field of mathematics that made a lot of scientific progress right before creation of OWL. I have no intention of belittling accomplishments of researchers in this field. However, there is little connection between this research and the practical data modeling needs of the common real world software systems. — [16]

thereby suggesting that SHACL and DLs are two completely separated worlds and as such contradicting the introductory paragraphs of this paper. On top of that, SHACL is presented by some stakeholders [21] as an alternative to the Web ontology language OWL [13], which is based on the description logic SROIQ [7].

This naturally begs the question: which misunderstanding is it that drives this wedge between communities? How can we explain this discrepancy from a mathematical perspective (thereby patently ignoring strategic, economic, social, and other aspects that play a role).

3 SHACL, OWL, and Description Logics

Our answer is that there are two important differences between OWL and SHACL that deserve attention. These differences, however, do not contradict the central thesis of this paper, which is that *SHACL is a description logic.*

1. The first difference is that **in SHACL, the data graph (implicitly) represents a first-order interpretation, while in OWL, it represents a first-order theory (an ABox)**. Of course, viewing the same syntactic structure (an RDF graph) as an interpretation is very different from viewing it as a theory. While this is a discrepancy between OWL and SHACL, theories as well as interpretations exist in the world of description logic and as such, this view is perfectly compatible with our central thesis. There is, however, one caveat with this claim that deserves some attention, and that is highlighted by the use of the world "implicitly". Namely, to the best of our knowledge, it is never mentioned that the data graph simply represents a standard first-order interpretation, and it has not been made formal what *exactly* the interpretation is that is associated to a graph. Instead, SHACL's language features are typically evaluated *directly* on the data graph. There are several reasons why we believe it is important to make this translation of a graph into an interpretation *explicit*.
   - This translation makes *the assumptions SHACL makes about the data* explicit. For instance, it is often informally stated that "SHACL uses closed-world assumptions" [10]; we will make this statement more precise: SHACL uses closed-world assumptions with respect to the relations, but open-world assumptions on the domain.
   - Once the graph is eliminated, we are in familiar territory. In the field of description logics a plethora of language features have been studied. It now becomes clear how to add them to SHACL, if desired. The 20+ years of research mentioned in [16] suddenly become directly applicable to SHACL.
2. The second difference, which closely relates to the first, is that **OWL and SHACL have a different (default) inference task**: the standard inference task at hand in OWL is *deduction*, while in SHACL, the main task is validation of RDF graphs against shape schemas. In logical terminology, this is evaluating whether a given interpretation satisfies a theory (TBox), i.e., this is the task of *model checking*.
   Of course, the fact that a different inference task is typically associated with these languages does not mean that their logical foundations are substantially different. Furthermore, recently, other researchers [11,14,15] have started to investigate tasks such as *satisfiability* and *containment* (which are among the tasks typically studied in DLs) for SHACL, making it all the more obvious that the field of description logics has something to offer for studying properties of SHACL.

In the next section, we develop our formalization of SHACL, building on the work mentioned above. Our formalization differs form existing formalizations of SHACL in a couple of small but important ways. First, as we mentioned, we explicitly make use of a first-order interpretation, rather than a graph, thereby indeed showing that SHACL is in fact a description logic. Second, the semantics for SHACL we develop would be called a "natural" semantics in database theory [1]: variables always range over the universe of all possible nodes. The use

of the natural semantics avoids an anomaly that crops up in the definitions of Andreşel et al. [2], where an "active-domain" semantics is adopted instead, in which variables range only over the set of nodes actually occurring in the input graph. Unfortunately, such a semantics does not work well with constants. The problem is that a constant mentioned in a shape may or may not actually occur in the input graph. As a result, the semantics adopted by Andreşel et al. violates familiar logic laws like De Morgan's law. This is troublesome, since automated tools (and humans!) that generate and manipulate logic formulas may reasonably and unwittingly assume these laws to hold. Also other research papers (see Remark 4) contain flaws related to not taking into account nodes that *do not* occur in the graph. This highlights the importance of taking a logical perspective on SHACL.

A minor caveat with the natural semantics is that decidability of validation is no longer totally obvious, since the universe of nodes is infinite. A solution to this problem is well-known from relational databases [1, Theorem 5.6.1]. Essentially, an application of solving the first-order theory of equality, one can reduce, over finite graphs, an infinite domain to a finite domain, by adding symbolic constants [3,8]. It turns out that in our case, just a single extra constant suffices.

In this paper, we will not give a complete syntactic translation of SHACL shapes to logical expressions. In fact, such a translation has already been developed by Corman et al. [6], and was later extended to account for all SHACL features by Jakubowski [9]. Instead, we show very precisely how the data graph at hand can be viewed as an interpretation, and that after this small but crucial step, we are on familiar grounds and know well how to evaluate expressions.

4 SHACL: The Logical Perspective

In this section of the paper we begin with the formal development. We define shapes, shape schemas, and validation. Our point of departure is the treatment by Andreşel et al. [2], which we adapt and extend to our purposes.

From the outset we assume three disjoint, infinite universes N, S, and P of *node names*, *shape names*, and *property names*, respectively.[3] We define *path expressions* E and *shapes* ϕ by the following grammar:

$$E ::= p \mid p^- \mid E \cup E \mid E \circ E \mid E^* \mid E?$$
$$\phi ::= \top \mid s \mid \{c\} \mid \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \geq_n E.\phi \mid eq(p, E) \mid disj(p, E) \mid closed(Q)$$

where p, s, and c stand for property names, shape names, and node names, respectively, n stands for nonzero natural numbers, and Q stands for finite sets of property names. In description logic terminology, a node name c is a *constant*, a shape name is a *concept name* and a property name is a *role name*.

[3] In practice, node names, shape names, and property names are IRIs [17], hence the disjointness assumption does not hold. However, this assumption is only made for simplicity of notation.

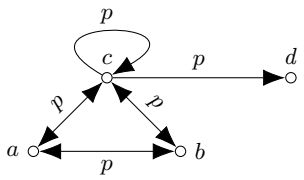


Fig. 1. An example graph to illustrate language features of SHACL.

As we will formalize below, every property/role name evaluates to a binary relation, as does each path expression. In the path expressions, p^- represents the inverse relation of p, $E \circ E$ represents composition of binary relations, E^* the reflexive-transitive closure of E and $E?$ the reflexive closure of E. As we will see, shapes (which represent unary predicates) will evaluate to a subset of the domain. The three last expressions are probably the least familiar. Equality ($eq(p, E)$) means that there are outgoing p-edges (edges labeled p) exactly to those nodes for which there is a path satisfying the expression E (defined below) . Disjointness ($disj(p, E)$) means that there are *no* outgoing p-edges to which there is also a path satisfying E. For instance in the graph in Figure 1, $eq(p, p^*)$ would evaluate to $\{c\}$, since c is the only node that has direct outgoing p-edge to all nodes that are reachable using only p-edges, and $disj(p, p^-)$ would evaluate to $\{d\}$ since d is the only node that has no symmetric p-edges. Closedness is also a typical SHACL feature: $closed(Q)$ represents that there are no outgoing edges about any predicates other than those in Q. In our example figure $closed(\{p\})$ would evaluate to $\{a, b, c, d\}$ and $closed(\{q\})$ to the empty set.

Remark 1. Andreşel et al. [2] also have the construct $\forall E.\phi$, which can be omitted (at least for theoretical purposes) as it is equivalent to $\neg \geq_1 E.\neg\phi$. In our semantics, the same applies to $\phi_1 \wedge \phi_2$ and $\phi_1 \vee \phi_2$, of which we need only one as the other is then expressible via De Morgan's laws. However, here we keep both for the sake of our later Remark 3. In addition to the constructors of Andreşel et al. [2], we also have $E?$, *disj*, and *closed*, corresponding to SHACL features that were not included there. □

A *vocabulary* Σ is a subset of $N \cup S \cup P$. A path expression or shape is said to be *over* Σ if it only uses symbols from Σ. On the most general logical level, shapes are evaluated in *interpretations*. We recall the familiar definition: An interpretation I over Σ consists of

1. a set Δ^I, called the *domain* of I;
2. for each constant $c \in \Sigma$, an element $[\![c]\!]^I \in \Delta^I$;
3. for each shape name $s \in \Sigma$, a subset $[\![s]\!]^I$ of Δ^I; and
4. for each property name $p \in \Sigma$, a binary relation $[\![p]\!]^I$ on Δ^I.

On any interpretation I as above, every path expression E over Σ evaluates to a binary relation $[\![E]\!]^I$ on Δ^I, and every shape ϕ over Σ evaluates to a subset of Δ^I, as defined in Tables 1 and 2.

E	$[\![E]\!]^I$
p^-	$\{(a,b) \mid (b,a) \in [\![p]\!]^I\}$
$E_1 \cup E_2$	$[\![E_1]\!]^I \cup [\![E_2]\!]^I$
$E_1 \circ E_2$	$\{(a,b) \mid \exists c : (a,c) \in [\![E_1]\!]^I \wedge (c,b) \in [\![E_2]\!]^I\}$
E^*	the reflexive-transitive closure of $[\![E]\!]^I$
$E?$	$[\![E]\!]^I \cup \{(a,a) \mid a \in \Delta^I\}$

Table 1. Semantics of a path expression E in an interpretation I over Σ.

ϕ	$[\![\phi]\!]^I$
$\top$	Δ^I
$\{c\}$	$\{c^I\}$
$\phi_1 \wedge \phi_2$	$[\![\phi_1]\!]^I \cap [\![\phi_2]\!]^I$
$\phi_1 \vee \phi_2$	$[\![\phi_1]\!]^I \cup [\![\phi_2]\!]^I$
$\neg\phi_1$	$\Delta^I \setminus [\![\phi_1]\!]^I$
$\geq_n E.\phi_1$	$\{a \in \Delta^I \mid \sharp([\![\phi_1]\!]^I \cap [\![E]\!]^I(a)) \geq n\}$
$eq(p,E)$	$\{a \in \Delta^I \mid [\![p]\!]^I(a) = [\![E]\!]^I(a)\}$
$disj(p,E)$	$\{a \in \Delta^I \mid [\![p]\!]^I(a) \cap [\![E]\!]^I(a) = \emptyset\}$
$closed(Q)$	$\{a \mid [\![p]\!]^I(a) = \emptyset$ for every $p \in \Sigma \setminus Q\}$

Table 2. Semantics of a shape ϕ in an interpretation I over Σ. For a set X, we use $\sharp X$ to denote its cardinality. For a binary relation R and an element a, we use $R(a)$ to denote the set $\{b \mid (a,b) \in R\}$.

As argued above, we define a *shape schema* $\mathcal{S}$ over Σ as a tuple (D,T), where

- D is an *acyclic TBox* [4], i.e., a finite set of expressions of the form $s \equiv \phi_s$ with s a shape name in Σ and ϕ_s a shape over Σ and where
  1. each s occurs exactly once as the left-hand-side of such an expression and
  2. the transitive closure of the relation $\{(s,t) \mid t$ occurs in $\phi_s\}$ is acyclic.
- T is a TBox, i.e., a finite set of statements of the form $\phi_1 \sqsubseteq \phi_2$, with ϕ_1 and ϕ_2 shapes.

If $\mathcal{S} = (D,T)$ is a shape schema over Σ and I an interpretation over $\Sigma \setminus S$, then there is a unique interpretation $I \diamond D$ that agrees with I outside of S and that satisfies D, i.e., such that for every expression $s \equiv \phi_s \in D$, $[\![s]\!]^{I\diamond D} = [\![\phi_s]\!]^{I\diamond D}$. We say that I *conforms to* $\mathcal{S}$, denoted by $I \models \mathcal{S}$, if $[\![\phi_1]\!]^{I\diamond D}$ is a subset of $[\![\phi_2]\!]^{I\diamond D}$, for every statement $\phi_1 \sqsubseteq \phi_2$ in T. In other words, I conforms to $\mathcal{S}$ if there exists an interpretation that satisfies $D \cup T$ that coincides with I on $N \cup P$.

Remark 2. In real SHACL, a shape schema is called a "shapes graph". There are some notable differences between shapes graphs and our shape schemas.

First, we take abstraction of some features of real SHACL, such as checking data types like numbers and strings.

Second, in real SHACL, the left-hand side of an inclusion statement in T is called a "target" and is actually restricted to shapes of the following forms:

a constant ("node target"); $\exists r.\{c\}$ ("class-based target", where r is 'rdf:type'); $\exists r.\top$ ("subjects-of target"); or $\exists r^-.\top$ ("objects-of target"). Our claims remain valid if this syntactic restriction imposed.

Third, in real SHACL not every shape name needs to occur in the left-hand side of a defining rule. The default that is taken in real SHACL is that shapes without a definition are *always satisified*. On the logical level, this means that for every shape s name that has no explicit definition, a definition $s \equiv \top$ is implicitly assumed. □

5 From Graphs to Interpretations

Up to this point, we have discussed the logical semantics of SHACL, i.e., how to evaluate a SHACL expression in a standard first-order interpretation. However, in practice, SHACL is not evaluated on interpretations but on RDF graphs. In this section, we show precisely and unambiguously how to go from a graph to a logical interpretation (in such a way that the actual SHACL semantics coincides with what we described above). A *graph* is a finite set of *facts*, where a fact is of the form $p(a, b)$, with p a property name and a and b node names. We refer to the node names appearing in a graph G simply as the *nodes* of G; the set of nodes of G is denoted by N_G. A pair (a, b) with $p(a, b) \in G$ is referred to as an *edge*, or a *p-edge*, in G. The set of p-edges in G is denoted by $[\![p]\!]^G$ (this set might be empty).

We want to be able to evaluate *any* shape on *any* graph (independently of the vocabulary the shape is over). Thereto, we will unambiguously associate, to any given graph G, an interpretation I over $N \cup P$ as follows:

- Δ^I equals N (the universe of all node names).
- $[\![c]\!]^I$ equals c itself, for every node name c.
- $[\![p]\!]^I$ equals $[\![p]\!]^G$, for every property name p.

If I is the interpretation associated to G, we use $[\![E]\!]^G$ and $[\![\phi]\!]^G$ to mean $[\![E]\!]^I$ and $[\![\phi]\!]^I$, respectively.

Remark 3. Andreşel et al. [2] define $[\![\phi]\!]^G$ a bit differently. For a constant c, they define $[\![\{c\}]\!]^G = \{c\}$ like we do. For all other constructs, however, they define $[\![\phi]\!]^G$ to be $[\![\phi]\!]^I$, but with the domain of I taken to be N_G, rather than N. In that approach, if $c \notin N_G$, $[\![\neg\neg\{c\}]\!]^G$ would be empty rather than $\{c\}$ as one would expect. For another illustration, still assuming $c \notin N_G$, $[\![\neg(\neg\phi \wedge \neg\{c\})]\!]^G$ would be $[\![\phi]\!]^G$ rather than $[\![\phi]\!]^G \cup \{c\}$, so De Morgan's law would fail. We tested both of these examples with existing SHACL implementations and all of these implementations indeed coincide with our semantics. Details of this (executable with actual SHACL engines) are included in Appendix A. □

Remark 4. The use of active domain semantics has also introduced some errors in previous work. For instance [11, Theorem 1] is factually incorrect. The problem originates with the notion of *faithful assignment* introduced by Corman et al. [6]

and adopted by Leinberger et al. This notion is defined in an active-domain fashion, only considering nodes actually appearing in the graph. For a concrete counterexample to that theorem, consider a single shape named s defined as $\exists r.\top$, with target $\{b\}$. In our terminology, this means that

$$D = \{s \equiv \exists r.\top\}, \text{ and}$$
$$T = \{\{b\} \sqsubseteq s\}.$$

On a graph G in which b does not appear, we can assign $\{s\}$ to all nodes from G with an outgoing r-edge (meaning that all these nodes satisfy s and no other shape (names)), and assign the empty set to all other nodes (meaning that all other nodes do not satisfy any shape). According to the definition, this is a faithful assignment. However, the inclusion $\{b\} \sqsubseteq s$ is not satisfied in the interpretation they construct from this assignment, thus violating their Theorem 1. □

The bug in [11], as well as the violation of De Morgan's laws will only occur in corner cases where the shape schema mentions nodes that not occur in the graph. After personal communications, Leinberger et al. [11] included an errata section where they suggest to fix this by demanding that (in order to conform) the target queries do not mention any nodes not in the graph. While technically, this indeed resolves the issue. Under that condition, Theorem 1 indeed holds, this solution in itself has weaknesses as well. Indeed, shape schemas are designed to validate graphs not known at design-time, and it should be possible to check conformance of *any* graph with respect to *any* shape schema. As the following example shows, it makes sense that a graph should conform to a schema in case a certain node does *not* occur in the graph (or does not occur in a certain context), and that — contrary to the existing SHACL formalizations — the natural semantics indeed coincides with the behaviour of SHACL validators in such cases.

Example 1. Consider a schema with $D = \emptyset$ and T consisting of a single inclusion

$$\{LuisLeiva\} \sqsubseteq \neg\exists(author \circ venue).\{BNAICBNLEARN2021\},$$

which states that a BNAIC PC chair does not author any BNAIC paper. If Luis Leiva does not occur in the list of of accepted papers, this list should clearly[4] conform to this schema. In all the proposed active domain semantics, however, the answer will be negative.

The definition of I makes — completely independent of the actual language features of SHACL — a couple of assumptions explicit. First of all, SHACL uses unique names assumptions (UNA): each constant is interpreted in I as a different domain element. Secondly, if $p(a, b)$ does not occur in the graph, it is assumed to be *false*. However, if a node c does not occur anywhere in the graph, it is not assumed to not exist. The domain of I is infinite! Rephrasing this: SHACL makes the Closed World Assumption on predicates, but not on objects.

[4] Technically, the standard is slightly ambiguous with respect to nodes not occurring in the data graph, but the behaviour of all existing validators (see AppendixA) corresponds to what is "clearly" the correct behaviour here.

Effective evaluation Since the interpretation defined from a graph has the infinite domain N, it is not immediately clear that shapes can be effectively evaluated over graphs. As indicated above, however, we can reduce to a finite interpretation. Let $\Sigma \subseteq N \cup P$ be a finite vocabulary, let ϕ be a shape over Σ, and let G be a graph. From G we define the interpretation $I_\star$ over Σ just like I above, except that the domain of $I_\star$ is not N but rather

$$N_G \cup (\Sigma \cap N) \cup \{\star\},$$

where $\star$ is an element not in N. We use $[\![\phi]\!]^G_\star$ to denote $[\![\phi]\!]^{I_\star}$ and find:

Theorem 1. *For every $x \in N_G \cup (\Sigma \cap N)$, we have $x \in [\![\phi]\!]^G$ if and only if $x \in [\![\phi]\!]^G_\star$. For all other node names x, we have $x \in [\![\phi]\!]^G$ if and only if $\star \in [\![\phi]\!]^G_\star$. Hence, I conforms to $\mathcal{S}$ if and only if $I_\star$ does.*

Theorem 1 shows that conformance can be performed by finite model checking, but other tasks typically studied in DLs are not decidable; this can be shown with a small modification of the proof of undecidability of the description logic $\mathcal{ALRC}$, as detailed by Schmidt-Schauß [18].

Theorem 2. *Consistency of a shape schema (i.e., the question whether or not some I conforms to $\mathcal{S}$) is undecidable.*

Following description logic traditions, decidable fragments of SHACL have been studied already; for instance Leinberger et al. [11] disallow equality, disjointness, and closedness in shapes, as well as union and Kleene star in path expressions.

6 Related Work and Conclusion

Formal investigations of SHACL have started only relatively recently. We already mentioned the important and influential works by Corman et al. [6] and by Andreşel et al. [2], which formed the starting point for the present paper. The focus of these papers is mainly on the extending the semantics to *recursive* SHACL schemas, which are not present in the standard yet, and which we also do not consider.

The connection between SHACL and description logics has also been observed by two other groups of researchers [11,14,15]. There, the focus is on typical reasoning tasks from DLs applied to shapes, and on reductions of these tasks to decidable description logics or decidable fragments of first-order logic. In its most general form, this cannot work (see Theorem 2), but the addressed works impose restrictions on the allowed shape expressions.

Next to shapes, other proposals for adding integrity constraints to the semantic web have been proposed, for instance by integrating them in OWL ontologies [20,12]. There, the entire ontology is viewed as an incomplete database.

None of the discussed works takes the explicit viewpoint that a data graph represents a standard first-order interpretation or that SHACL validation is model checking. We took this viewpoint and in doing so formalized precisely

how SHACL relates to the field of description logics. There are (at least) three reasons why this formalization is important. First, it establishes a bridge between two communities, thereby allowing to exploit the many years of research in DLs also for studying SHACL. Second, our formalization of SHACL clearly separates two orthogonal concerns:

1. *Which information does a data graph represent?* This is handled in the translation of a graph into its *natural interpretation*.
2. *What is the semantics of language constructs?* This is handled purely in the well-studied logical setting.

Third, as we showed above, our formalization corresponds closer to actual SHACL than existing formalizations, respects well-known laws (such as De Morgan's) and avoids issues with nodes not occurring in the graph requiring special treatment. As such, we believe that by rooting SHACL in the logical setting, we have devised solid foundations for future studies and extensions of the language.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Andreşel, M., Corman, J., Ortiz, M., Reutter, J., Savkovic, O., Simkus, M.: Stable model semantics for recursive SHACL. In: Proceedings of WWW. pp. 1570–1580 (2020)
3. Aylamazyan, A., Gilula, M., Stolboushkin, A., Schwartz, G.: Reduction of the relational model with infinite domains to the case of finite domains. Doklady Akademii Nauk SSSR **286**(2), 308–311 (1986), in Russian
4. Baader, F., Calvanese, D., McGuiness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook. Cambridge University Press (2003)
5. Boneva, I., Gayo, J.L., Prud'hommeaux, E.: Semantics and validation of shape schemas for RDF. In: Proceedings of ISWC. pp. 104–120 (2017)
6. Corman, J., Reutter, J., Savkovic, O.: Semantics and validation of recursive SHACL. In: Proceedings of ISWC. pp. 318–336 (2018), extended version, technical report KRDB18-01
7. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible SROIQ. In: Proceedings of KR. pp. 57–67 (2016)
8. Hull, R., Su, J.: Domain independence and the relational calculus. Acta Informatica **31**, 513–524 (1994)
9. Jakubowski, M.: Formalization of SHACL. `https://www.mjakubowski.info/files/shacl.pdf`, accessed: 2021-06-16
10. Knublauch, H.: SHACL and OWL compared. `https://spinrdf.org/shacl-and-owl.html`, accessed: 2021-06-16
11. Leinberger, M., Seifer, P., et al.: Deciding SHACL shape containment through description logics reasoning. In: Proceedings of ISWC. pp. 366–383 (2020)
12. Motik, B., Horrocks, I., Sattler, U.: Bridging the gap between OWL and relational databases. J. Web Semant. **7**(2), 74–89 (2009)
13. OWL 2 Web ontology language: Structural specification and functional-style syntax. W3C Recommendation (Dec 2012)

14. Pareti, P., Konstantinidis, G., et al.: SHACL satisfiability and containment. In: Proceedings of ISWC. pp. 474–493 (2020)
15. Pareti, P., Konstantinidis, G., Mogavero, F.: Satisfiability and containment of recursive SHACL. CoRR **abs/2108.13063** (2021), `https://arxiv.org/abs/2108.13063`
16. Polikoff, I.: Why I don't use OWL anymore – Top Quadrant blog. `https://www.topquadrant.com/owl-blog/`, accessed: 2021-06-04
17. RDF 1.1 primer. W3C Working Group Note (Jun 2014)
18. Schmidt-Schauß, M.: Subsumption in KL-ONE is undecidable. In: Proceedings of KR. pp. 421–431 (1989)
19. Shapes constraint language (SHACL). W3C Recommendation (Jul 2017)
20. Tao, J., Sirin, E., Bao, J., McGuinness, D.L.: Integrity constraints in OWL. In: Proceedings of AAAI (2010)
21. TopQuadrant: An overview of SHACL: A new W3C standard for data validation and modeling. `https://www.topquadrant.com/an-overview-of-shacl/` (2017), webinar slides

A Actual SHACL specifications

In this appendix, we provide some actual SHACL specifications that support our claims that for some, possibly more controversial, choices our semantics indeed corresponds to actual SHACL. All the examples presented below have been tested on three SHACL implementations: Apache Jena SHACL[5] (using their Java library) TopBraid SHACL[6] (using their Java library as well as their online playground), and Zazuko[7] (using their online playground).

All the examples in this section will assume the following prefixes are defined:

```
@prefix : <http://www.example.org/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
```

Example 2. The following SHACL shape `:MyShape` states that the node `:c` must have at least one outgoing `:r`-edge.

```
:MyShape a sh:NodeShape ;
    sh:property [
        sh:path :r ;
        sh:minCount 1 ] .

:MyShape sh:targetNode :c .
```

When validating a graph that does not contain the node `:c` against `:MyShape`, it will return a violation stating that the node `:c` does not have an `:r`-edge. This supports the choice made in [2], as well as in our work, to evaluate constants (node names) to themselves, even if they do not occur in the data graph. When translating this into the logical setting, this example thus shows that **all node names, even those that do not occur in the graph, are part of the domain**, which is exactly how our natural interpretation is defined.

In our formal notation, this shapes graph corresponds to the shape schema

$$\begin{aligned} &:\texttt{MyShape} \equiv \exists :\texttt{r}.\top \\ &\{:\texttt{c}\} \sqsubseteq :\texttt{MyShape}, \end{aligned}$$

where the first line is the definition of `:MyShape`, and the second line its target. □

Example 3. The following SHACL shape `:MyShape` states that all nodes with an `:r`-edge must conform to the `:NoDef` and `:AlsoNoDef` shapes which we do not define.

```
:MyShape a sh:NodeShape ;
    sh:and ( :NoDef :AlsoNoDef ) .

:MyShape sh:targetSubjectsOf :r .
```

[5] https://jena.apache.org/documentation/shacl/index.html
[6] https://shacl.org/playground/
[7] https://shacl-playground.zazuko.com/

When validation a graph containing only the triple `:a :b :r` , it will validate without violation. This supports our observation that **shapes without an explicit definition are assumed to be satisfied by all nodes (i.e., are interpreted as $\top$)**.

In our formal notation, this shapes graph corresponds to the shape schema

$$\texttt{:MyShape} \equiv \texttt{:NoDef} \wedge \texttt{:AlsoNoDef}$$
$$\exists \texttt{:r}.\top \sqsubseteq \texttt{:MyShape}$$

where again, the first line is the definition of `:MyShape`, and the second line its target. □

Example 4. The following SHACL shape `:MyShape` states that it cannot be so that the node `:this` is different from itself (i.e., that it must be equal to itself, but specified with a double negation).

```
:MyShape a sh:NodeShape ;
    sh:not [ sh:not [ sh:hasValue :this ] ] .

:MyShape sh:targetNode :this .
```

Clearly, this shape should validate every graph and it does so in all SHACL implementations we tested. This supports our **choice of the natural semantics**, rather than the active domain semantics of [2] (see also Remark 3). Indeed, in that semantics, this shape will never validate any graph because the left-hand side of the inclusion will be evaluated to be the empty set.

The inclusions

$$\texttt{:MyShape} \equiv \neg\neg\{\texttt{:this}\}$$
$$\{\texttt{:this}\} \sqsubseteq \texttt{:MyShape}$$

again formalize the above shapes graph. □

Example 5. Another example in the same vein as the previous, to show that the **natural semantics** correctly formalizes is the one where [2]'s semantics does not respect the De Morgan's laws (again, see Remark 3), as follows:

```
:MyShape a sh:NodeShape ;
    sh:not [
        sh:and (
            [ sh:not [
                  sh:path :r ;
                  sh:minCount 1 ] ]
            [ sh:not [ sh:hasValue :this ] ] ) ] .

:MyShape sh:targetNode :this .
```

This shape graph becomes

$$:\texttt{MyShape} \equiv \neg(\neg\exists:\texttt{r}.\top \wedge \neg\{:\texttt{this}\})$$
$$\{:\texttt{this}\} \sqsubseteq :\texttt{MyShape}$$

when translated to our formalism. □

Example 6. Finally, in the following SHACL shapes graph, the shape `:NotAnAuthor` holds for all nodes (whether or not they occur in the data graph) that are not an author of a BNAIC paper. This example illustrates the **utility of targeting nodes that do no occur in the graph**.

```
:NotAnAuthor a sh:NodeShape ;
    sh:not [
        a sh:PropertyShape ;
        sh:path (:author :venue) ;
        sh:qualifiedValueShape [ sh:hasValue :BNAICBNLEARN2021 ] ;
        sh:qualifiedMinCount 1 ] .

:NotAnAuthor sh:targetNode :LuisLeiva .
```

which corresponds to

$$\{\mathit{LuisLeiva}\} \sqsubseteq \neg\exists(\mathit{author} \circ \mathit{venue}).\{\mathit{BNAICBNLEARN2021}\},$$

from Example 1.

When the node `:LuisLeiva` does not occur in the data graph, then in every SHACL implementation, this schema validates: any graph that does not contain the node `:LuisLeiva` (or where that node is not an author of a BNAICBNLEARN2021 paper), conforms to the above shapes graph, supporting our argument in Example 1. □

Explainable and Interpretable Features of Emotion in Human Body Expressions

André Mertens[1,2][0000−0003−1807−7770], Esam Ghaleb[1,3][0000−0002−0603−9817], and Stylianos Asteriadis[1,4][0000−0002−4298−6870]

[1] Department of Data Science and Knowledge Engineering, Maastricht University, The Netherlands
[2] andre.mertens@student.maastrichtuniversity.nl
[3] esam.ghaleb@maastrichtuniversity.nl
[4] stelios.asteriadis@maastrichtuniversity.nl

Abstract. The cooperation between machines and humans could be improved if machines could understand and respond to the emotions of the people around them. Furthermore, the features that machines use to classify emotions should be explainable to reduce the inhibition threshold for automatic emotion recognition. However, the explainability in bodily expressivity of emotions has hardly been explored yet. Therefore, this study aims to visualize and explain the features used by neural networks to classify emotions based on body movements and postures of human characters in videos. For this purpose, a state-of-the-art neural network was selected as classification model. This network was used to classify the videos of two datasets for emotion classification. As a result, the activation of the classification features used by the model were visualized with heatmaps over the course of the videos. Furthermore, a combination of Class Activation Maps and body joint coordinates were used to compute the activation of body parts in order to investigate the existence of prototypical activation patterns in emotions. As a result, similarities were found between the activation patterns of the two datasets. These patterns may provide new insights into the classification features used by neural networks and the emotion expression in body movements and postures.

Keywords: emotion recognition · bodily emotional expressions · deep learning · explainable AI · XAI

1 Introduction

Many future machine applications in the areas of care, education, and social robotics, among others, will require close collaboration between machines and humans. That is why machines used in these areas, in particular, can benefit from a comprehensive understanding of the people around them. Emotional state recognition can provide a more natural human-machine interaction, with machines responding to people's actions according to their emotional state. The

emotion classification studies use different types of models to distinguish emotions from each other such as categorical [6] and dimensional [15] models. The categorical model, which is utilized in this study, divides emotions into several different categories. For example, Ekman et al. categorizes emotional expressions in six basic emotions, which contain *Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*, and *Surprise* [6]. The categorical model simplifies the emotional state recognition task and transforms it into a classification task. Thus, an approach to emotion recognition does not need to capture the emotional state entirely, but rather needs to learn features to classify emotions accordingly.

The state-of-the-art approaches focus predominantly on facial expressions to classify emotions [4] [5]. However, the use of facial expressions alone can often lead to ambiguity of emotion. Besides, there may be applications for emotion recognition where the human body is present, but the face may be distant, hidden, or obscured [8]. Body expressions can be utilized in emotion recognition since they encode rich information about the emotional state of a person [14]. Thus, in recent studies, both body and facial expressions are used to classify emotions [12]. For example, Filntisis et al. [7] presented a method to combine children's body posture and facial expressions for emotion classification. Their results showed that the additional use of body expressions could significantly improve emotion recognition.

Moreover, in recent years, it has become essential not only to create neural networks with high accuracy, but also to develop methods that provide insight into the computed classification features of the neural networks. Neural networks do not give direct information about the features used in classification since they are black-box models [17]. Although explainable AI is currently getting attention in recent studies [9] [17], explainability in bodily expressivity is still hardly explored.

To address these issues, this study aims to visualize and explain the features used by neural networks to classify emotions based on body expressions of human characters in videos. For this purpose, a neural network was trained to classify emotions based on two video datasets. Subsequently, the classification features learned by the neural network were visualized with heatmaps to analyze them qualitatively. There is a lack of a quantitative approach to compare the features used by 3D-Convolutional Neural Networks to classify emotions. Thus, in this study, a method was developed to compare the activation of specific body parts for different emotions.

2 Related Work

In the presentation of the Body Language Dataset (BoLD) (section 3.1), Luo et al. [14] already tested different approaches for classifying the dataset. A distinction was made between the approaches learning from skeleton and learning from pixels. For the learning from skeleton approach, the body key points (section 3.5) of the persons to be classified were determined. These body key points were used to distinguish the emotions portrayed in the videos using classification models.

For this purpose, 2 different methods were presented for learning from skeletons. With the first method, handcrafted features were analyzed in the videos using Laban Movement Analysis (LMA) [13]. Subsequently, these features were used as input to a Random Forest Classifier to classify the videos. The second method is based on the fact that the movement of the key points over the course of the video can be represented as a spatiotemporal graph. Therefore, Spatial Temporal Graph Convolutional Networks (ST-GCN) were used as an end-to-end feature learning method to classify the videos based on the body key points. In contrast, classification features were determined directly in the RGB frames of the videos for the learning from pixels approach. For the learning from pixels approach, also 2 different methods were presented. For the first method, Support Vector Machines (SVM) were used to classify the videos based on trajectory based handcrafted features. In contrast, the performance of different 2D- and 3D-Convolutional Neural Networks (CNN) was validated on the dataset for the second method. Luo et al. found that 2D- and 3D-CNNs perform significantly better than all the other models on the BoLD dataset [14]. One of the best performances was delivered by a Two-Stream Inflated 3D ConvNet (I3D) [2]. The I3D model (section 3.3) is a 3D-CNN architecture that uses video sequences as input. Thus, the model learns spatiotemporal classification features directly from the RGB frames of the video sequences.

Hiley et al. [9] summarized the state-of-the-art methods for explainable deep learning on video classification tasks. They highlighted that application of deep neural networks on video sequences for tasks like action and emotion recognition is currently at the forefront of computer vision. As a result, they mentioned that a wide variety of work is devoted to this task. However, they note the lack of research on explanations for these methods. Nevertheless, Hiley et al. [9] emphasized the Saliency Tubes [17] method as particularly promising. For this method, Class Activation Maps (CAMs) of a neural network are computed for the complete duration of the video. This not only makes it possible to identify regions in each frame where classification features are present. Furthermore, this method also provides the ability to display frames where these features are higher concentrated [17].

In their study, Dael et al. [3] investigated the extent to which patterns can be identified in actors' body movements when portraying emotions. It was shown that most emotions were systematically represented by several different body movement patterns. Only a few emotions were characterized by a single specific pattern. Thus, it was possible to differentiate the emotions. These patterns included different specific movements of body parts like the head, the arms, or the knees. Moreover, complete body movements were considered as a component in the patterns [3].

3 Methodology

Fig. 1 shows the proposed methodology to visualize and explain features used by neural networks for the classification of emotions. First, the neural network (here,

the I3D [2] model) is trained to classify the emotions presented in videos. The learned classification features are utilized to compute Class Activation Maps (CAMs) for the complete duration of the videos using the Saliency Tube [17] method. These CAMs are visualized using heatmaps. Besides, the body joint coordinates in the videos are obtained with OpenPose [1]. Finally, the combination of CAMs and body joint coordinates is utilized to validate the activation of certain body parts during the expression of emotions. The individual components of the proposed methodology are described in the following sections.

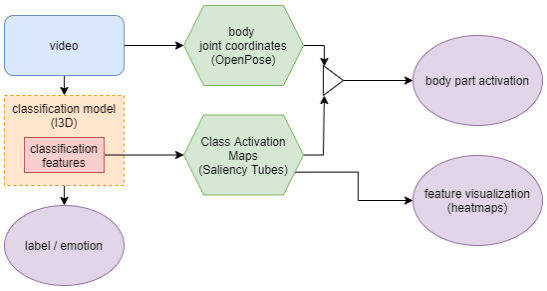


Fig. 1: Methodology flow chart

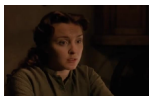
3.1 Datasets

The GreenStimuli dataset was developed by the Faculty of Psychology and Neuroscience (FPN) at Maastricht University [16]. This dataset was created to study body expression in the representation of different emotions. For this purpose, 871 videos were taken of persons depicting the 6 Ekmanian emotions (*Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*, *Surprise*) and *Neutral* . Fig. 2a illustrates that all persons wore long dark clothes and were recorded against a green background. Besides, the complete body was always recorded. Moreover, the faces were subsequently blurred to verify the representation of emotions independent of facial expressions.

Luo et al. [14] introduced the large-scale Body Language Dataset (BoLD) used in the Bodily Expressed Emotion Understanding (BEEU) challenge. The dataset consists of 9,876 videos recorded in the wild. Moreover, the videos may contain multiple persons expressing their emotions with body movements. This results in a total of 13,239 annotations created through crowdsourcing. The annotations include 26 categorical emotion labels and are split into training and validation subsets by the authors [14]. Fig. 2b and Fig. 2c display two example frames of the BoLD dataset. The field sizes can vary considerably since the dataset contains videos from the wild. Thus, the dataset includes videos as long shots (Fig. 2b) but also as close-ups (Fig. 2c).

(a) GreenStimuli frame [16]

(b) BoLD long shot [14]



(c) BoLD close-up [14]

Fig. 2: Dataset sample frames

3.2 Data preprocessing

The frame sizes of the videos are different not only among the datasets but also within the datasets. Therefore, the videos of both datasets were uniformly reduced to a width and height of 224×224 to use them as input for the I3D (section 3.3) model. Moreover, the videos were uniformly reduced to a number of 48 consecutive frames.

The GreenStimuli dataset does not have a train, validation, and test subset split, hence, the dataset was split into training (70%, 605), validation (15%, 131), and testing (15%, 135) subsets. Furthermore, no label balancing method was performed for the GreenStimuli dataset since the video amount across the labels were similar.

It is not in the scope of this study to run an extensive analysis for too many emotions. Instead, this study focuses on the most basic emotions to explore the potential of explaining emotions using body expressions. Therefore, 5 basic emotions (*Anger*, *Fear*, *Happiness*, *Sadness*, *Surprise*) were selected from the 26 categorical emotions of the BoLD dataset for the classification task. These 5 basic emotions correspond to 5 of the 7 labels of the GreenStimuli dataset. There are no corresponding labels for *Disgust* and *Neutral* in the BoLD dataset, resulting in different number of classes for each dataset.

Originally, the videos of the BoLD dataset were labeled with a float value in the range of 0 to 1 for each emotion. Thus, the magnitude of the float value is used to represent how much a label applies to an expressed emotion in a video. To use the BoLD dataset in the scope of this study, the classification task was converted into a single-label classification. Therefore, it was searched for videos that were maximal for one of the 5 basic emotions. If the value was maximum for one of the basic emotions, this basic emotion was selected and assigned to the corresponding video as a single label. Videos were discarded if the videos were not maximal for one of the basic emotions or maximal for several basic emotions. There were 2030 videos for training and 268 videos for validation after the BoLD dataset was converted to a single-label classification task. The training and validation subsets were unbalanced. Therefore, a weighted loss function and accuracy score depending on the occurrence of the labels was used to handle the imbalance of the BoLD dataset. Moreover, the person whose emotion is to be classified was cropped from the videos since there can be multiple people simultaneously in the videos of the BoLD dataset.

3.3 Model structure

The Two-Stream Inflated 3D ConvNet (I3D) [2] model is a 3D-CNN that performs classification based on video sequences. Through the 3D-convolutional layers, the I3D model learns spatial and temporal classification features directly from the RGB frames. Carreira et al. [2] pretrained the I3D model on the DeepMind Kinetics [10] dataset. The Kinetics dataset contains humans performing 400 different actions. Besides, the I3D model has been successfully used for emotion classification [14]. The best performance on the Kinetics dataset was obtained by training the model separately on an RGB and an optical flow field stream of the videos and averaging the outputs for testing [2]. Optical flow fields were omitted in the scope of this study since this would have made it difficult to conclude about the detected features and the associated explanatory power of the features. Besides, the original architecture of the I3D model could be preserved mostly as presented by the original authors. Fig. 3 presents that the layers of the original I3D model were taken over from the first convolutional layer to the ninth and last Inception Module. An Inception Module is a combination of convolutional and pooling layers with their outputs concatenated into a single output vector. The last Inception Module was originally followed by a final layer sequence containing an average pooling and a 3D-convolutional layer as classification layer. This final layer sequence was replaced by a global average pooling layer and a fully connected linear layer as classification layer. The size of the output vector was adjusted to the label amount of the respective dataset. The GreenStimuli dataset has 7 labels, hence, the output vector corresponds to a size of (7×1). In contrast, the vector has a size of (5×1) for the BoLD dataset since only 5 labels were classified.

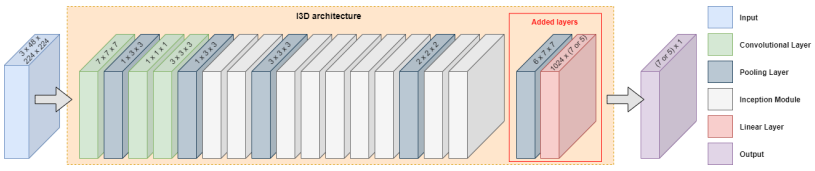


Fig. 3: Model architecture

3.4 Classification feature visualization

Saliency Tube [17] is a method to visualize classification features of 3D-CNNs for a class of interest. This is intended to provide human-interpretable visual explanations for various classification problems. For this purpose, Saliency Tubes display regions of the input pixels that contain classification features at frame level and over time. Therefore, Saliency Tubes are an extension of the Class Activation Mapping (CAM) [18] method for visualizing activation maps of 2D-CNNs. Saliency Tubes are computed using equation 1 where a is the weights of

the classification layer, F is the convolution features of the last convolutional layer, i is the selected label index, and j is the feature index.

$$saliency\_tube_i = \sum_{j=1} a_{i,j} \cdot F_j \tag{1}$$

First, all weights $a_{i,j}$ are multiplied by the corresponding convolution feature F_j. A convolutional feature is a matrix of the size $(n' \times h' \times w')$. n' corresponds to the frame amount of the convolution feature, h' corresponds to the height, and w' to the width of a convolution feature frame, hence, h' and w' are the two spatial dimensions and n' is the temporal dimension. Therefore, a float value $(a_{i,j})$ is multiplied by a matrix (F_j). Convolution features with higher weights get higher intensity and convolution features with lower weights get lower intensity. The intensity of the weighted convolution features, thus, indicates how strongly they contribute to the classification. This intensity of weighted convolution features is referred to as activation. The activation matrices of all convolution features are summed up to consider the entire activation of convolution features. The result is a matrix also with the dimensions $(n' \times h' \times w')$ that contains the information of all convolution features cumulatively. This matrix is called Saliency Tube. The frames of Saliency Tubes along the temporal dimension are called Class Activation Maps [17]. Furthermore, the Saliency Tube is reshaped to match the original video dimensions $(n \times h \times w)$ by using the spline interpolation to increase the spatiotemporal dimensions (equation 2). Consequently, Saliency Tube matrix has one activation value per pixel for the original video. The values were normalized in a range from 0 to 1 since the value ranges of Saliency Tubes can vary considerably from each other [17].

$$saliency\_tube_i = (n' \times h' \times w') \xrightarrow[factors=(\frac{n}{n'} \times \frac{h}{h'} \times \frac{w}{w'})]{interpolation} (n \times h \times w) \tag{2}$$

The final Class Activation Maps of Saliency Tubes can be visualized with heatmaps. Therefore, the heatmaps have a gradient from red for important features to blue for unimportant features [17]. Fig. 4 presents heatmaps in the course of a GreenStimuli dataset video, which is labeled as *Happiness*. The red-colored regions in Fig. 4b show that classification features for this video are contained in the arms and the upper body. In contrast, the legs are colored blue and thus, do not contain classification features. Besides, arms and torso are colored relatively small-scale green/yellow at the beginning (Fig. 4a) and end (Fig. 4c) of the video. In contrast, the arms and torso are colored red over a large area in the middle of the video sequence (Fig. 4b). This shows that the classification features seem to cluster in the middle of the video sequence. At the beginning and end of the video, they are relatively more weakly represented.

3.5 Calculation of the body part activation

Dael et al. [3] presented the connection of emotions with the movement of certain body parts. Therefore, the extent to which the I3D model use classification

(a) Start frame heatmap (b) Middle frame heatmap (c) End frame heatmap

Fig. 4: Heatmaps in the course of a sample video

features of arms, legs, and the torso should be further explored. For this purpose, the OpenPose [1] library was used to estimate body key points of people in the videos. In this study, the COCO key point format that contains 18 body key points was utilized [1]. The key points can be used to calculate the coordinates of the body joints. Therefore, the coordinates of the points that lie on a line between two key points were computed. Then, the body joints were divided into 3 categories (arm: green, leg: yellow, torso: purple) to generalize the calculation of body part activation. The combination of Class Activation Maps (section 3.4) and body joint coordinates can be used to calculate the activation of the different body parts in the videos. Therefore, the average activation values for torso, arm, and leg joints were obtained separately for each video. Fig. 5 presents a) a sample frame of a GreenStimuli dataset video with the obtained heatmap, b) 17 body joints extracted from the same frame, and c) the combination of the extracted joints and heatmap.

(a) Heatmap sample (b) Body joint sample (c) Joints with heatmap

Fig. 5: Heatmap and joint samples

4 Results

This chapter covers the results of the proposed methodology on the two datasets. In the following sections, the model performance is evaluated on the datasets, classification features are visualized, and the activation of the individual body parts is calculated for different emotions.

4.1 Model performance evaluation

The model was trained for 100 epochs with a batch size of 32 using the cross-entropy loss function and optimized with Adam [11] where the learning rate was

10^{-3}. After each epoch, the validation dataset was used to measure the general accuracy of the models. After training, the model with the highest validation accuracy was selected. The I3D model achieved a training accuracy of 72.6% and 39.8%, and a validation accuracy of 64.1% and 34.2% for the GreenStimuli and BoLD dataset, respectively. For the GreenStimuli dataset, the performance of the I3D model was additionally evaluated with the test dataset. The model achieved a test accuracy of 68.9%.

4.2 Classification feature visualization

Fig. 6a and 6b show heatmap samples for two labels of the GreenStimuli dataset. Based on the red colored regions, it is visualized that especially for *Anger* the arms seem to contain the classification features. For *Fear*, the complete body seems to contain classification features. In Fig. 6c and 6d, heatmap samples are displayed for two labels of the BoLD dataset. For *Anger*, the faces seem to contain the most important classification features. The postures of the torso and especially the hands seem to include classification features for *Fear*.

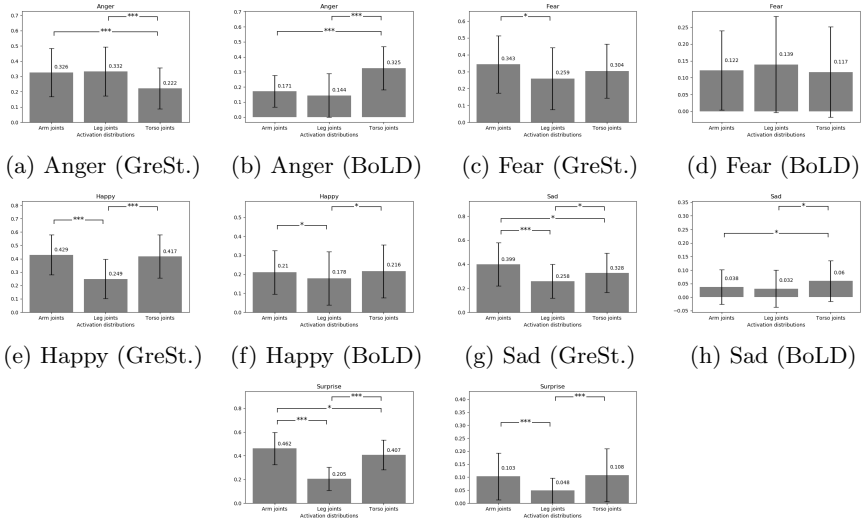
(a) Anger (GreSt.) (b) Fear (GreSt.) (c) Anger (BoLD) (d) Fear (BoLD)

Fig. 6: Heatmap samples for both datasets

4.3 Calculation of the body part activation

The 3 activation values for arm, leg, and torso joints were calculated as described in section 3.5 for all videos. Then, only the activation values were selected of the videos that were correctly predicted by the model in order to discard features or activation values that led to incorrect predictions. Consequently, distributions of activation values for all 3 joint categories of all emotions were obtained. In Fig. 7, the activation of the different body parts is shown for the 5 common labels of the datasets. The distributions of the 3 joint categories were plotted as a bar plot with average (bar) and standard deviation (antennas) for the different labels. Furthermore, Welch's t-test was performed to test the similarity of the different distributions where a p-value less than 0.05 indicates a significant difference. For this purpose, the difference was categorized as $p < 0.05$ for slightly significant (*), $p < 0.001$ for considerably significant (**), and $p < 0.0001$ strongly significant (***). These significant differences are additionally displayed in the bar plot diagrams. The main similarities between the joint activation for the respective labels of both datasets are listed below:

- In Fig. 7a and Fig. 7b, similarities can be identified between the activation patterns for *Anger*. The activation of the arm and leg joints is not significantly different from each other for both datasets.
- A similar activation pattern can also be detected for *Fear* (Fig. 7c and Fig. 7d). The activation of the torso joints is not significant different from the activation of the arm and leg joints for both datasets.
- It is illustrated that the activation patterns for *Happiness* for both datasets behave identically (Fig. 7e and Fig. 7f). For both datasets, there are two levels of activation. The leg joints have the lowest activation significantly. Arm and trunk joint activations are not significantly different from each other. However, they are at a higher activation level than leg joints.
- For *Sadness* (Fig. 7g and Fig. 7h), no common pattern can be identified between the two datasets.
- Also for *Surprise*, a similarity between the activation pattern is shown for the GreenStimuli (Fig. 7i) and the BoLD dataset (Fig. 7j). The leg joint activation is strongly significantly the lowest for both datasets.



(a) Anger (GreSt.) (b) Anger (BoLD) (c) Fear (GreSt.) (d) Fear (BoLD)

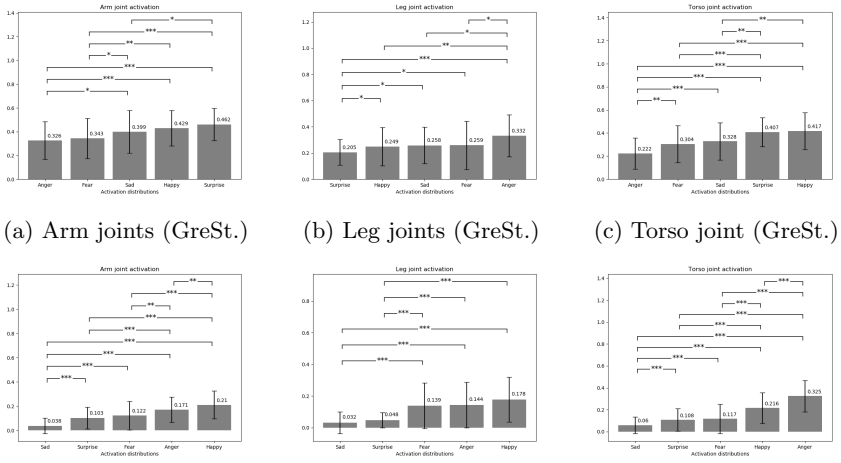
(e) Happy (GreSt.) (f) Happy (BoLD) (g) Sad (GreSt.) (h) Sad (BoLD)

(i) Surprise (GreSt.) (j) Surprise (BoLD)

Fig. 7: Activation of the joint categories for the respective labels

In Fig. 8, the activation of the respective joint category is displayed for the 5 common labels of the datasets. It is illustrated that the activation patterns behave significantly differently for both datasets. The similarities between the activation of a certain joint category for different labels of both datasets are listed below:

– *Happiness* is at the highest activation level for the arm joint activation for both datasets (Fig 8a and Fig. 8d).
– The commonalities of the activation patterns of the two datasets for the leg joints (Fig. 8b and Fig. 8e) are that the activation for *Surprise* is on the lowest activation level. *Anger*, on the other hand, is on the highest activation level.
– For the torso joint activation (Fig. 8c and Fig. 8f), it can only be roughly said that the torso joint activation for *Happiness* is on high activation levels for both datasets. *Happiness* is at the highest activation level for the Green-Stimuli dataset. In contrast, it is at the second highest activation level for the BoLD dataset.



(a) Arm joints (GreSt.) (b) Leg joints (GreSt.) (c) Torso joint (GreSt.)

(d) Arm joint (BoLD) (e) Leg joint (BoLD) (f) Torso joint (BoLD)

Fig. 8: Activation of a certain joint category for different labels

5 Discussion

5.1 Model performance evaluation

The performance of the I3D model on the GreenStimuli dataset is significantly better than the performance on the BoLD dataset. Luo et al. also achieved only an average precision of 15.37% with an I3D model on all 26 labels of the BoLD dataset [14]. This average precision value can not be compared with the accuracy scores from the experiments of this study due to the label reduction and the different preprocessing steps. However, the precision value shows that it is a hard

dataset to classify in general. Relatively low accuracy scores of a classification model can be assumed since the basic conditions of the classification of the BoLD dataset are already difficult.

5.2 Classification feature visualization

In section 4.2, clear differences can be observed between the obtained heatmaps of the models trained on both datasets. In the GreenStimuli dataset videos, the model always has all body parts available. On the other hand, in the BoLD dataset, the legs of the subjects in the videos are often not present. Furthermore, the videos of the GreenStimuli dataset are similar in their presentation, which allowed the model to be fully trained on the actual classification of emotions. With the BoLD dataset, the model additionally had to learn to deal with different field sizes (section 3.1). This makes the classification task for the model even more difficult. Moreover, it is displayed that the model used classification features in faces, especially in the close-ups and medium shots of the BoLD dataset videos. This is a major difference from the GreenStimuli dataset, as this information is not available at all due to the blurred faces. Although the I3D models have the same architecture and classified similar labels, they used different classification features. The reason for this is the different nature of the two datasets.

5.3 Calculation of the body part activation

The basic conditions (different field sizes and face representations) of the two datasets differ considerably, and therefore both models learned partly distinctly different classification features (section 5.2). Nevertheless, the activation of the body parts for the 5 common labels of the datasets was compared to assess whether similar activation patterns can be identified despite the large differences (section 4.3). Despite the different nature of the two datasets, similarities can be found in the activations of the body parts. However, the patterns found can at best be declared as prototype patterns. To test the activation patterns for generalizability, applying the proposed methodology to more video datasets for emotion classification is necessary.

6 Limitations

Although the presented method seems to provide measurable and comparable activation patterns, this method has limitations. For example, there are some cases where the model correctly predicts the video label. However, the computed Class Activation Maps do not give any information about the learned classification features. A related GreenStimuli dataset sample is displayed in Fig. 9a. In this case, hardly any features are used of the person's body for emotion classification. Moreover, the main classification features tend to be detected outside the body. Unfortunately, it is not clear from these example what features the model used to classify this video.

The BoLD dataset presents different problems since it contains videos obtained in the wild. There are events or elements in videos that the model uses as classification features that are not directly related to the portrayed emotion. Two examples are shown in Fig. 9b and Fig. 9c. In figure 9b, it is displayed that playing a guitar is used as a classification feature for *Happiness* in some videos. The reason for this is that videos of characters playing guitar predominantly appear for the label *Happiness*. In most cases, this may also be true that playing an instrument is associated with a positive emotion. However, playing an instrument can not be used as a clear indicator for *Happiness*. Furthermore, despite the person to be classified is cropped from the videos, it can not be prevented that other people are visible in the image details. These, in turn, can also lead to side effects. In Fig. 9c, the person to be classified is displayed in the front of the video. However, the model uses the dancing people in the background to classify the emotion.

(a) Hardly body features (b) Playing guitar (c) Additional persons

Fig. 9: Samples for limitations of the method

Despite the limitations described above, the Class Activation Maps overwhelmingly provide explainable information about what body part features were used to classify the emotions. Therefore, the samples with unexplained features or side effects were also used to calculate the activation of the joint categories (section 4.3) for the sake of completeness. This means that the Class Activation Maps were selected automatically without any manual intervention. Therefore, the samples loaded with the more unexplained features or side effects could influence the determined activation patterns. These samples, thus, often formed the outliers of the statistics as their activation values differed from the activation values of samples with body-related features. This explains the partially high standard deviations in the bar charts (Fig. 7 and Fig. 8). However, since the majority of the Class Activation Maps contain explainable body-related features, it was assumed that the activation values and activation patterns, on average, can be attributed to body parts.

7 Conclusion

In this study, a new method is presented to calculate and compare the influence of different body parts on the expression of emotions. For this purpose, I3D was identified as a model that is suitable for the classification of emotions in

videos since the I3D model is capable of providing classification features that can be explained in most cases. Heatmaps were used to visualize the classification features obtained from the I3D model. Then, by combining the Class Activation Maps and the body joint coordinates, the activation of certain body parts for the different emotions was calculated. Finally, activation patterns were identified for each emotion based on the similarity or difference of joint activations.

The proposed methodology was applied on the GreenStimuli and BoLD datasets. Some of the patterns in the activations were identical or at least quite similar for both datasets, although the two datasets differ in their nature. Nevertheless, these patterns may only be considered as prototype patterns since they still need to be verified on further video datasets for emotion classification to be considered as general patterns.

In future studies, the presented method for calculating the activation of body parts could be improved by increasing the accuracy of the classification model through different architectures or by using different hyper-parameters in training. Furthermore, an automatic selection method could be developed to classify Class Activation Maps as explainable and unexplainable, which would lead to the definition of thresholds for the activation values of the body joints. A selection of the Class Activation Maps could reduce the standard deviation of the calculated body joint activation values. Besides, the calculation of body joint activation could be made much more fine-grained to determine the activation not only for torso, arm, and leg joints but also for the head, hand, and foot joints separately.

To the best of our knowledge, there is no quantitative approach to compare the features used by 3D-CNNs in the classification of emotions. Moreover, the link between emotion expression, and body gestures and movements is hardly explored. This study aims to provide new insights into the classification features used by 3D-CNNs and the emotion expression in body movements and postures.

Acknowledgement

This study has been supported by the Faculty of Psychology and Neuroscience at Maastricht University.

References

1. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(1), 172–186 (12 2018), http://arxiv.org/abs/1812.08008
2. Carreira, J., Zisserman, A.: Quo Vadis, action recognition? A new model and the kinetics dataset. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. vol. 2017-Janua, pp. 4724–4733. Institute of Electrical and Electronics Engineers Inc. (5 2017). https://doi.org/10.1109/CVPR.2017.502, http://arxiv.org/abs/1705.07750
3. Dael, N., Mortillaro, M., Scherer, K.R.: Emotion expression in body action and posture. Emotion **12**(5), 1085–1101 (2012). https://doi.org/10.1037/a0025737

4. De Gelder, B.: Why bodies? Twelve reasons for including bodily expressions in affective neuroscience (2009). https://doi.org/10.1098/rstb.2009.0190
5. Du, S., Tao, Y., Martinez, A.M.: Compound facial expressions of emotion. Proceedings of the National Academy of Sciences of the United States of America **111**(15) (2014). https://doi.org/10.1073/pnas.1322355111
6. Ekman, P., Friesen, W.V.: A new pan-cultural facial expression of emotion. Motivation and Emotion **10**(2), 159–168 (1986). https://doi.org/10.1007/BF00992253
7. Filntisis, P.P., Efthymiou, N., Koutras, P., Potamianos, G., Maragos, P.: Fusing Body Posture With Facial Expressions for Joint Recognition of Affect in Child–Robot Interaction. IEEE Robotics and Automation Letters **4**(4), 4011–4018 (2019). https://doi.org/10.1109/lra.2019.2930434
8. Filntisis, P.P., Efthymiou, N., Potamianos, G., Maragos, P.: Emotion Understanding in Videos Through Body, Context, and Visual-Semantic Embedding Loss. In: Computer Vision – ECCV 2020 Workshops, pp. 747–755. Springer International Publishing (10 2020). https://doi.org/10.1007/978-3-030-66415-2_52, http://arxiv.org/abs/2010.16396
9. Hiley, L., Preece, A., Hicks, Y.: Explainable deep learning for video recognition tasks: A framework & recommendations (9 2019), http://arxiv.org/abs/1909.05667
10. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The Kinetics Human Action Video Dataset (5 2017), http://arxiv.org/abs/1705.06950
11. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. International Conference on Learning Representations, ICLR (12 2015), https://arxiv.org/abs/1412.6980v5
12. Kleinsmith, A., Bianchi-Berthouze, N.: Affective body expression perception and recognition: A survey. IEEE Transactions on Affective Computing **4**(1), 15–33 (2013). https://doi.org/10.1109/T-AFFC.2012.16
13. Laban, R.v., Ullmann, L.: The Mastery of Movement. ERIC (1971)
14. Luo, Y., Ye, J., Adams, R.B., Li, J., Newman, M.G., Wang, J.Z.: ARBEE: Towards Automated Recognition of Bodily Expression of Emotion in the Wild. International Journal of Computer Vision **128**(1) (8 2020). https://doi.org/10.1007/s11263-019-01215-y, http://arxiv.org/abs/1808.09568 http://dx.doi.org/10.1007/s11263-019-01215-y
15. Mehrabian, A.: Pleasure-Arousal-Dominance: A general framework for describing and measuring individual differences in temperament. Current Psychology **14**(4), 261–292 (1996). https://doi.org/10.1007/bf02686918
16. Poyo Solanas, M., Vaessen, M.J., de Gelder, B.: The role of computational and subjective features in emotional body expressions. Scientific Reports **10**(1) (2020). https://doi.org/10.1038/s41598-020-63125-1
17. Stergiou, A., Kapidis, G., Kalliatakis, G., Chrysoulas, C., Veltkamp, R., Poppe, R.: Saliency Tubes: Visual Explanations for Spatio-Temporal Convolutions. In: Proceedings - International Conference on Image Processing, ICIP. vol. 2019-Septe, pp. 1830–1834. IEEE Computer Society (9 2019). https://doi.org/10.1109/ICIP.2019.8803153, https://youtu.be/JANUqoMc3es
18. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 2016-Decem, pp. 2921–2929. IEEE Computer Society (12 2016). https://doi.org/10.1109/CVPR.2016.319, http://arxiv.org/abs/1512.04150

Deep Learning Techniques for Detection and Diagnosis of Brain Metastases

Mariia Pliusnova and Alexia Briassouli[0000−0002−0545−3215]

Department of Data Science and Artificial Intelligence
Maastricht University

Abstract. As a requirement for brain tumor diagnosis and therapy, accurate and dependable diagnosis of brain tumors based on Magnetic Resonance Images (MRI) is required. The automated detection and segmentation of tumors may take place using deep learning methods, which are increasingly being used for medical image diagnostics due to their high accuracy. This work examines the use of two deep learning approaches for detection and segmentation: both detection (CNN) and segmentation (U-Net) models were designed and trained on the benchmarking OASIS and BraTS 2020 datasets. On the test dataset, the detection model obtained the accuracy of 0.946 and the segmentation model obtained the accuracy of 0.994. The evaluation of segmenation model for the whole tumor (WT), the tumor core (TC) and the enhancing tumor (ET) achieved dice coefficients of 0.85, 0.74, 0.67, respectively. These results are equivalent to the currently published state-of-the-art, but are twice as fast on average. Despite being relatively simple, the proposed strategy has resulted in a good and balanced performance and can be a valuable diagnostics tool for doctors. The proposed solution is openly available at https://github.com/grimjjow/`Medical_Image_Analysis`

Keywords: Brain tumor detection · CNN · U-Net · Tumor segmentation

1 Introduction, Motivation

Magnetic resonance imaging (MRI) is widely used for diagnosing and evaluating brain tumors[20], as it provides distinctive tissue contrast. The manual inspection of MRI brain tumor images is a time-consuming process, highly dependent on the doctor's level of expertise, and may produce varied and subjective results. Today various computer vision algorithms are being developed for the automated analysis of brain tumor MRI images, achieving increasingly high accuracy[23]. The automated detection of tumors helps doctors localize them more efficiently, while their segmentation supports the classification of important tumor regions.

Deep Learning (DL) techniques are currently gaining ground, as they outperform traditional machine learning techniques[24]. Specifically, Convolutional neural networks(CNNs) dominated the field during the ImageNet Large-Scale Visual Recognition Challenge(ILSVRC)[17] and proved their ability to accurately detect and localize different types of objects. Benchmarking datasets for

brain tumor imaging have been made public for fundamental problems of brain tumor diagnostics, such as detection and segmentation. In this work, the OASIS-3 dataset images[10] and images from the Brain Tumor Segmentation (BraTS) Challenge 2020 are used, which provides real-world data of MRI brain tumor scans[11]. Models with improving accuracy have been proposed for these challenges, however, they are often time-consuming, and tailor-made to specific problems, and datasets, which make them non-generalizable[19].

The main motivation of this work is to explore and develop simple but generally applicable methods that can accurately detect and segment tumors with high accuracy, with very limited computational resources. Due to limited time and memory resources, the datasets used in this work are not as large as some that are used in high-level SoA. Moreover, the architectures used are designed to be not very deep and are simply constructed, aiming to provide a general detection solution for tumors.

2 Datasets for Brain Tumor Detection and Segmentation

2.1 Detection Challenge

MRI images from the OASIS-3 dataset(Cross-Sectional and Longitudinal)[10] were obtained for training and testing. There are 3000 images of T1 and T2-weighted MRI images, which are classified in two categories: healthy patients and patients with tumors. Figure 1 depicts several OASIS brain MRIs.

Fig. 1: Example of OASIS MRI images in two categories: yes respresents an image containing a tumor, no represents an image not containing a tumor.

2.2 Segmentation Challenge

The BraTS 2020 dataset contains 369 training, 125 validation and 166 test multimodal brain MRI studies[2][3][11]. Each study has four MRI images: T1-weighted (T1), post-contrast T1-weighted (T1ce), T2-weighted (T2), and fluid attenuated inversion recovery (Flair) sequences[22]. All these sequences were used for both training and testing. Four distinct tumoral subregions can be defined from the

images: the “Enhancing Tumor” (ET), the “Non-Enhancing Tumor” (NET), the “Necrotic tumor” (NCR) and the “Peritumoral Edema” (ED)[22]. These subregions can be then combined to classify three main tumoral regions. The ET is the first region. The ET with the addition of NET and NCR creates the ”Tumor Core”(TC) region. Finally, the ED with the TC creates the ”Whole Tumor”(WT). This subregion classification is used in many SoA approaches that work with BraTS datasets[8]. Examples of each region are provided in Fig. 2.

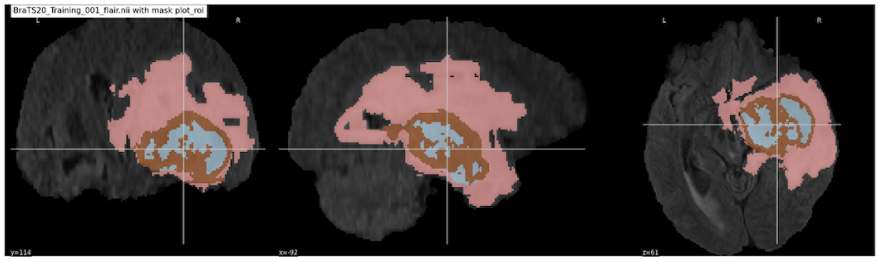


Fig. 2: Example of an MRI image of a brain tumor from the BraTS 2020 dataset. Brown represents an enhancing tumor (ET), Blue represents a non-enhancing tumor/necrotic tumor (NET/NCR), and Pink represents peritumoral edema (ED).

3 Methods

To develop robust brain tumor detection and segmentation, a combination of various image processing and deep learning techniques was applied. The initial data underwent preprocessing, and was then used for training and testing.

Data Preprocessing: Preprocessing is an important first step, even in automated computer-based diagnostics, as MRI images may accumulate noise, for example, due to patient motion. For this reason, images are first cropped, then resized and normalized. Normalization is done by identifying the largest contour of the brain in every image and cropping the images according to an outline of the contour points of the brain (Figure 3). Data augmentation is also applied, to increase variation in the training data, and thus improve robustness and accuracy, using the Keras ImageDataGenerator. It includes horizontal and vertical flip, rotation, shift, zoom, and brightness adjustment. These transformations are chosen as they result in slight distortion of original data, but do not create completely different shapes, which can negatively affect the model evaluation.

3.1 Detection Convolutional Neural Network

As a part of the primary analysis of the MRI images, a CNN was designed to detect the tumors and classify images according to the outcomes of the detection

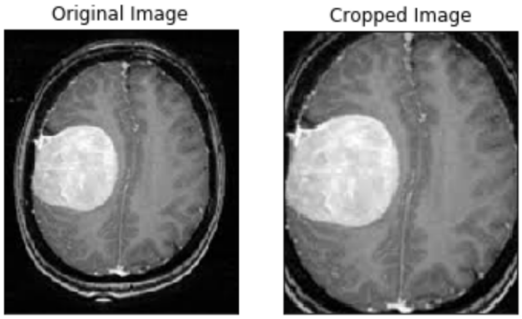


Fig. 3: Image preprocessing: initial image, normalized and cropped image.

classifier. The CNN architecture proposed in this paper (Figure 4), referred to in the sequel as DCNN, is built upon standard CNN models[14], with original optimization approaches. A 2D CNN is used for this purpose, although 3D solutions have also been developed. This is because one of the goals of this work is to find very low-resource solutions to accurate tumor detection. In the literature, 3D data is often separated into 2D slices, from which rich features are extracted by the CNN.

At a starting point, the model takes an input layer of images of a specified size (200 $\times$ 200) that are then relayed to the first processing block. This block consists of two convolution layers containing 32 feature kernel filters of 5 x 5 size. A new feature map of 200 $\times$ 200 $\times$ 32 dimensions is obtained and combined in the max-pooling layer built with a stride size of 2 pixels and 2 $\times$ 2 kernels. This procedure decreases the spatial dimension of the preceding layer's feature map to 100 x 100 x 64. Following the max-pooling layer, the output is routed to the second processing block composed of two convolution layers. It includes 64 feature kernel filters of 3 x 3 size. Then the updated feature map is again relayed to the max-pooling layer built with a stride size of 2 pixels and 2 $\times$ 2 kernels. The resulting combined map then has the spatial dimension of 50 $\times$ 50 $\times$ 128. This process is repeated in two more blocks. Finally, the last feature map is processed through two fully connected layers that contain the ReLU activation function.

3.2 U-Net Based Convolutional Neural Network

U-Net is one of the most popular CNN architectures designed for more complex level classification problems like medical image segmentation. The proposed U-Net model utilizes the idea of simplicity, meaning fewer layers and the use of 2D volumes instead of 3D. These design choices were made to lower time and memory costs, as a 2D U-Net model can process a full slice at once while a 3D model can only process a small patch of the 3D volume[6]. A modified architecture of a standard U-Net model was developed and is shown in Figure 5.

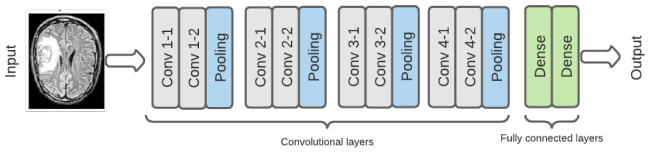


Fig. 4: Detection CNN model architecture

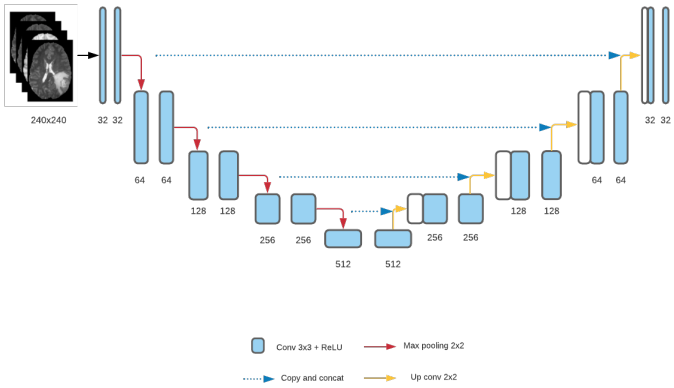


Fig. 5: U-Net model architecture

The encoder path is built using 5 processing blocks. Each block has two convolutional layers with a filter size of 3 × 3, a stride of 1, and rectifier activation. As such, these layers increase the number of feature maps from 1 to 512. The max-pooling layer with a stride of 2 × 2 is applied to every updated feature map in each block. Similarly, the decoder path also consists of 5 processing blocks. Every processing block starts with a deconvolutional layer with a filter size of 3 × 3 and a stride of 2 × 2. This effectively increases the size of feature maps in both directions while significantly reducing the number of feature maps. Finally, there are no fully connected layers invoked in the proposed model, which is very common in CNNs for classification problems[15].

4 Experiments

This section describes the evaluation of experiments and elaborates on the significance of the obtained results in the context of the research questions. The computational requirements of the methods described are very low, as experiments

were run locally on a CPU, demonstrating the effectiveness of deep learning approaches even in very resource-constrained environments

Training: In our experiments, both detection and segmentation models were trained with the dropout rate set to 0.2 using Adam Optimiser and learning rate of 0.001, which is proven to be an optimal approach for brain tumor diagnostics models[4]. The training consisted of 55 epochs with a batch size of 120 for the detection model and 35 epochs with a batch size of 250 for the segmentation model. A visualization of training process is shown for both detection and segmentation models in Figure 6 and Figure 7 respectively.

The proposed detection model obtained the best accuracy of 0.912 and a loss of 0.132. From Figure 6 it can be seen that the model slightly overfits due to the small dataset size. Although overfitting poses an issue, the model is still highly generalizable and is only constrained to the format of the input MRI images. This is achieved by applying data normalization and data augmentation, as well as using real-world MRI studies which have high variability.

The proposed segmentation model achieves a very high accuracy of 0.994, loss of 0.189, and a Dice coefficient of 0.656. Figure 7 shows that training and validation metrics are more consistent and stable, which shows that the U-Net architecture significantly improves the performance in comparison to a basic CNN model. This is expected since the U-Net model performs well in the SoA[24].

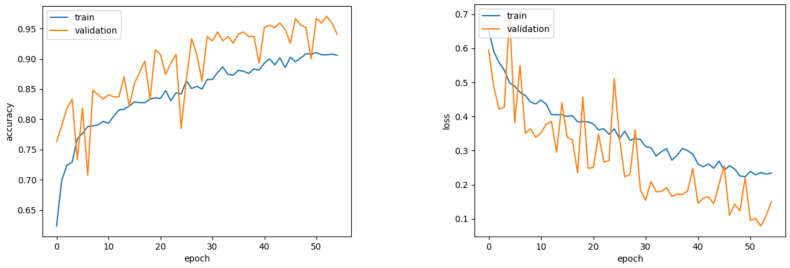


Fig. 6: Detection model training and validation: accuracy and loss

Testing: Detection

For detection, performance evaluation metrics are shown in Table 1. The proposed technique has a high accuracy of 0.946, as well as a notable F1-score(0.921) and precision value(0.923), all of which indicate the model's efficiency. Additionally, different configurations were added to the detection CNN model, starting with a basic model, as a control accuracy, and building upon it. These configurations, or improvements, are the following: a base model with Adam optimizer, an optimized model with Data augmentation, and optimized and augmented model with Dropout. These specific improvements were chosen based on findings in the related SoA. The various configurations and their corresponding accuracy values are shown in Table 2. While the base case already gives a relatively high accuracy(0.872), it is surprising to see that using different gradient

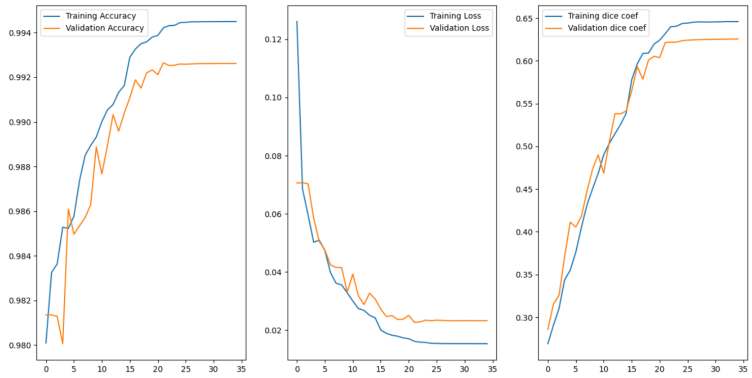


Fig. 7: Segmentation model training and validation: accuracy, loss and dice coef

descent optimizers, like the Adam optimizer, almost doesn't affect the accuracy. This issue was addressed via data augmentation and regularization in a form of a Dropout[7], which forces the model to learn important features independently. The final accuracy, which is obtained by adding Dropout, is considered as the best accuracy shown by the detection model. Finally, the proposed detection model is compared to other SoA algorithms in Table 3. Models that use standard classifiers like Random Forest[16] or Deep Neural Networks(DNN)[18] show accuracy below 90%, while models that utilize more modern approaches, like CNN, show higher accuracy, going up to a maximum of 95%. The proposed detection model obtains almost full 95% accuracy, which is a very good result considering that the training time of this model, approximately 1 hour, is a lot less than of some SoA, up to 17 hours[1].

Evaluation metrics	**Performance score**
Accuracy	0.946
Precision	0.923
Sensitivity	0.950
F1-score	0.921
ROC AUC	0.913

Table 1: Performance metrics of the DCNN model

Testing: Segmentation

For segmentation, the evaluation metrics in Table 4 indicate high accuracy(0.994) and high precision(0.994). Additionally, sensitivity(0.992) and mean IoU(0.831) values are calculated. All these values are higher than most SoA methods[24], so it can be stated that the proposed segmentation model has a high performance for the BraTS challenge.

CNN improvements	Accuracy
Base case	0.872
Adam optimizer	0.878
Data augmentation	0.913
Dropout	0.946

Table 2: Improvements and accuracy of various CNN optimizations

Method	Algorithm	Accuracy
Sobhaninia et al.[18]	Cascaded DNN	78.1%
Reza et al.[16]	Random Forest	86.7%
Nasim et al.[12]	SVM	92.4%
Proposed	DCNN	**94.6%**
Amin et al.[1]	2D CNN	95.1%

Table 3: Comparison of detection models

Table 5 shows the DSC results of our proposed segmentation model results for the whole tumor (WT), tumor core (TC) or edema (ED), and enhancing tumor (ET), respectively. Obtained performance scores were comparable to recently published studies within the scope of the BraTS challenge. Even though the studies overall contain the same type of data, the ones that are related to BraTS 2013 and BraTS 2015 datasets contain much fewer patient cases than the BRATS 2019 and BraTS 2020 datasets[5]. Taking this into account, a comparison of the methods was only done based on the testing data. When compared to the BraTS 2015 and BraTS 2019 challenge datasets, the suggested segmentation approach achieved equivalent results of 0.85 DSC for the WT segmentation. Segmentation of TC and ET showed higher performance for the older studies, but lower performance for the recent studies. This can be due to the fact that both these tumoral regions might look similar in some MRI images and this can cause an increase in accuracy loss.

5 Discussion, Conclusions

The proposed detection and segmentation models showed results comparable to the SoA methods, at a significantly lower computational cost, and provided valuable insights.

Evaluation metrics	Performance score
Accuracy	0.994
Precision	0.994
Sensitivity	0.992
Mean IoU	0.831

Table 4: Performance metrics of the U-Net model

Method	Data	WT	ED/TC	ET
Pereira et al.[13]	BraTS 2013 challenge	0.84	0.72	0.62
Havaei et al.[7]	BraTS 2015 challenge	0.79	0.58	0.69
Kamnitsas et al.[9]	BraTS 2015 challenge	0.85	0.67	0.63
Feifan et al.[21]	BraTS 2019 challenge	0.85	0.79	0.77
Proposed model	BraTS 2020 challenge	**0.85**	**0.74**	**0.67**
Henry et al.[8]	BraTS 2020 challenge	0.89	0.84	0.79

Table 5: Comparison of segmentation models

The evaluation of both proposed models (Tables 1, 4) shows all performance metrics' values are relatively high and the proposed models can be applied to real-world patient data, which indicates that the CNN architecture is an effective method for brain tumor detection and segmentation.

The approach proposed for detection showed higher accuracy than most of the traditional models and it required less computational power. Comparative analysis of segmentation approach was more complex, as it showed higher overall accuracy, but lower or equal accuracy per tumoral region. An important piece of evidence that affected the final segmentation model evaluation is that it has a computational time (per case) of half the time required on average for the SoA methods, which is a significant improvement. Summarizing all the observations provided the following answers:

The detection model has comparatively high performance compared to conventional methods and it can be expected to improve with increase and diversity to training and testing data. The segmentation model has a higher performance based on evaluation metrics values and computational time, excluding region-specific segmentation performance. This model can also improve by using target-centered training and testing methods, where targets are specific tumoral regions. Different configurations of the CNN model optimizations were tested and showed that regularization and data augmentation methods can be considered the most promising improvements in terms of increase of accuracy.

As a result of mentioned improvements, the proposed detection model achieved high performance values and it can be used in further research as a baseline architecture. For segmentation, while obtained results were not cutting-edge for the BraTS 2020 challenge, the proposed method's segmentation results are within the normal range of applicable models for tumor segmentation and could already be useful for clinical use.

Aside from assessing the accuracy and validity of brain tumor segmentation results, computation time is a significant factor. Having limited GPU and memory at hand, it wasn't possible to develop more advanced techniques for the models. On the other hand, the proposed segmentation model showed a computation time of approximately 14 seconds per case, which is up to the current standard computation time of a few minutes. Compared to this model some other studies were less computationally efficient with times of approximately 30 seconds[9], 20 seconds to 3 minutes[7] and 7 minutes[13] per case. All observa-

tions referring to computation time of both models shown significant benefits of proposed models compared to the SoA, which justifies the motivation and methodology used in this work.

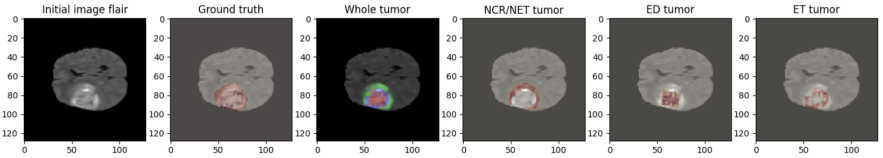


Fig. 8: Sample test image(1) segmentation obtained by proposed U-Net model

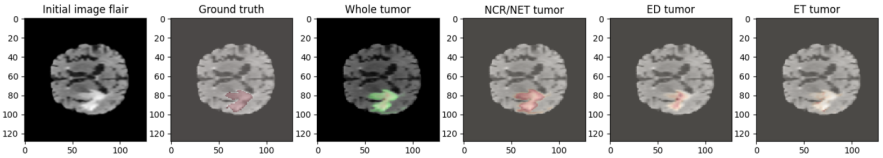


Fig. 9: Sample test image(2) segmentation obtained by proposed U-Net model

Figures 8 and 9 show segmentation results for the proposed U-Net model generated from the sample testing images. In both figures, the first image shows the initial flair, while the second image shows the manual ground truth segmentations. The results of the proposed automated approach for segmentation are shown in the next images. They include a whole tumor(WT), necrotic tumor(NET/NCR), tumor core(ED/TC), and enhancing tumor(ET) segmentations. By visual comparison of both figures, it can be noted that the segmentation of ET can be more complex for some cases, where the initial image contains various distortions(like noise) or the ET tissue is barely present in the MRI scan. Considering this it can be assumed that the ET segmentation is expected to have the lowest accuracy and it is in fact confirmed by the data in the results section. On the contrary, the segmentation of the WT is consistent and comparable to the manual segmentation.

In this work, computationally efficient and generalizable CNN and U-Net architectures are presented, which achieve SoA levels of brain tumor detection and segmentation. The simple architecture used in the CNN ensures it has a low computational cost and can be applied to a wider range of problems, as it is not tailor-made to a specific type of visual data. In future work, the two proposed models can be combined into an extensive specialized diagnostics system and in combination with solutions to the challenges listed above it can obtain even

better clinical results. This system could be a useful method for doctors to deliver a reliable medical diagnoses of brain tumors and help thousands of patients.

References

1. Amin, J., Sharif, M., Yasmin, M., Fernandes, S.: Big data analysis for brain tumor detection: Deep convolutional neural networks. Future Gener. Comput. Syst. 2018 **87**, 290–297 (2018). https://doi.org/https://doi.org/10.1016/j.future.2018.04.065
2. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., Freymann, J.B., Farahani, K., Davatzikos, C.: Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. Sci Data. 2017 Sep **4**, 170117 (2015). https://doi.org/https://doi.org/10.1038/sdata.2017.117
3. Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R.T.: Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. ArXiv **abs/1811.02629** (2019)
4. Bakr Siddique, M.A., Sakib, S., Rahman Khan, M.M., Tanzeem, A.K., Chowdhury, M., Yasmin, N.: Deep convolutional neural networks model-based brain tumor detection in brain mri images. Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 2020 pp. 909–914 (2020). https://doi.org/https://doi.org/10.1109/I-SMAC49090.2020.9243461
5. BraTS: Brain tumor segmentation (brats) challenge (2020), bluehttp://braintumorsegmentation.org
6. Dong, H., Yang, G., Liu, F., Mo, Y., Guo, Y.: Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. ArXiv **abs/1705.03820** (2017)
7. Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M., Larochelle, H.: Brain tumor segmentation with deep neural networks. Medical Image Analysis. 2017 Jan **35**, 18–31 (2017). https://doi.org/https://doi.org/10.1016/j.media.2016.05.004
8. Henry, T., Carre, A., Lerousseau, M., Estienne, T., Robert, C., Paragios, N., Deutsch, E.: Brain tumor segmentation with self-ensembled, deeply-supervised 3d u-net neural networks: a brats 2020 challenge solution. ArXiv **abs/2011.01045** (2020)
9. Kamnitsas, K., Ledig, C., Newcombe, V.F., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D., Glocker, B.: Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. Medical Image Analysis. 2017 Feb **36**, 61–78 (2017). https://doi.org/https://doi.org/10.1016/j.media.2016.10.004
10. LaMontagne, P.J., Benzinger, T.L., Morris, J.C., Keefe, S., Hornbeck, R., Xiong, C., Grant, E., Hassenstab, J., Moulder, K., Vlassenko, A., Raichle, M.E., Cruchaga, C., Marcus, D.: Oasis-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease (2019). https://doi.org/https://doi.org/10.1101/2019.12.13.19014902
11. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J.: The multimodal brain tumor image segmentation benchmark (brats). IEEE Trans Med Imaging. 2015 Oct **34**(10), 1993–2024 (2015). https://doi.org/https://doi.org/10.1109/TMI.2014.2377694
12. Nasim, M.A., Shah, F., Hossain, T., Ashraf, M., Shishir, F.: Brain tumor detection using convolutional neural network. Thesis. 2019 Jun (2019). https://doi.org/https://doi.org/10.13140/RG.2.2.15562.52163

13. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Brain tumor segmentation using convolutional neural networks in mri images. IEEE Transactions on Medical Imaging. 2016 **35**(5), 1240–1251 (2016). https://doi.org/https://doi.org/10.1109/TMI.2016.2538465
14. Qing, L., Weidong, C., Xiaogang, W., Yun, Z., Dagan, F.D., Mei, C.: Medical image classification with convolutional neural network. 13th International Conference on Control Automation Robotics Vision (ICARCV). 2014 pp. 844–848 (2014). https://doi.org/https://doi.org/10.1109/ICARCV.2014.7064414
15. Rehman, M.U., Cho, S., Kim, J.H., Chong, K.T.: Bu-net: Brain tumor segmentation using modified u-net architecture. Electronics. 2020 **9**(12), 2203 (2020). https://doi.org/https://doi.org/10.3390/electronics9122203
16. Reza, S.M.S., Mays, R., Iftekharuddin, K.M.: Multi-fractal detrended texture feature for brain tumor classification. Proc SPIE Int Soc Opt Eng. 2015 Feb **9414**, 941410 (2015). https://doi.org/https://doi.org/10.1117/12.2083596
17. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision volume. 2015 Apr **115**, 211–252 (2015). https://doi.org/https://doi.org/10.1007/s11263-015-0816-y
18. Sobhaninia, Z., Rezaei, S., Karimi, N., Emami, A., Samavi, S.: Brain tumor segmentation by cascaded deep neural networks using multiple image scales. ArXiv **abs/2002.01975** (2020)
19. Sobhaninia, Z., Rezaei, S., Noroozi, A., Ahmadi, M., Zarrabi, H., Karimi, N., Emami, A., Samavi, S.: Brain tumor segmentation using deep learning by type specific sorting of images. ArXiv **abs/1809.07786** (2018)
20. Villanueva-Meyer, J.E., Mabray, M.C., Cha, S.: Current clinical brain tumor imaging. Neurosurgery. 2017 Sep **81**(3), 397–415 (2017). https://doi.org/https://doi.org/10.1093/neuros/nyx103
21. Wang, F., Jiang, R., Zheng, L., Meng, C., Biswal, B.: 3d u-net based brain tumor segmentation and survival days prediction. Lecture Notes in Computer Science. 2020 p. 131–141 (2020). https://doi.org/https://doi.org/10.1007/978-3-030-46640-4.13
22. Wang, S., Dai, C., Mo, Y., Angelini, E., Guo, Y., Bai, W.: Automatic brain tumour segmentation and biophysics-guided survival prediction. ArXiv **abs/1911.08483** (2019)
23. Wu, W., Chen, A.Y.C., Zhao, L., Corso, J.J.: Brain tumor detection and segmentation in a crf (conditional random fields) framework with pixel-pairwise affinity and superpixel-level features. International Journal of Computer Assisted Radiology and Surgery volume. 2014 Mar **9**, 241–253 (2014). https://doi.org/https://doi.org/10.1007/s11548-013-0922-7
24. Zeineldin, R.A., Karar, M.E., Jan Coburger, C.R.W., Burgert, O.: Deepseg: deep neural network framework for automatic brain tumor segmentation using magnetic resonance flair images. International Journal of Computer Assisted Radiology and Surgery volume. 2020 Jun **15**, 909–920 (2020). https://doi.org/https://doi.org/10.1007/s11548-020-02186-z

ConveRT for FAQ Answering

Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans

CLiPS Research Center
Antwerp University, Belgium
firstname.lastname@uantwerpen.be

Abstract. Knowledgeable FAQ chatbots are a valuable resource to any organization. While powerful and efficient retrieval-based models exist for English, it is rarely the case for other languages for which the same amount of training data is not available. In this paper, we propose a novel pre-training procedure to adapt ConveRT, an English conversational retriever model, to other languages with less training data available. We apply it for the first time to the task of Dutch FAQ answering related to the COVID-19 vaccine. We show it performs better than an open-source alternative in both a low-data regime and a high-data regime.

Keywords: Chatbot · Conversational Agent · FAQ Answering · ConveRT · Transformers

1 Introduction

In this paper, we present a Dutch-based FAQ retrieval system trained using a limited amount of training data.

FAQ answering is the task of retrieving the right answer given a new user query. It is widely used in chatbots and has been studied for many years [6, 22, 9, 18, 10, 20], although the attention has shifted towards extractive question answering more recently [19], probably because of a lack of dedicated datasets. FAQ answering systems typically use retrieval systems [6, 22, 9, 18, 10, 20] rather than generative models grounded on external knowledge [13, 4, 14]. The generative approach is more flexible as it is able to generate new answers. However, these models suffer from knowledge hallucinations [21], limiting their usefulness in a corporate environment.

Most previous research focusing on FAQ retrieval and non-factoid question answering were developed for English. ConveRT [7], a response selection module available within Rasa [1], caught our attention as it is effective and does not require a GPU at inference time. Unfortunately, it is only available in English. Despite having significantly less conversational training data (400K pairs of utterances) than the original ConveRT model (727M pairs), we successfully trained the same model for Dutch.

Our contributions are the following:

– We show it is possible to train a ConveRT model for a non-English language using a limited number of conversation pairs by adopting a two-phase pre-training approach (general and conversational).

– We show that a Dutch ConveRT model performs better than the response selector module from Rasa, both in a low and high data regime.

2 Related Work

An FAQ dataset consists of pairs of questions and answers. The FAQ retrieval task involves ranking the available answers for a given user query. There are three methods available to solve this problem: matching a new user query on the available questions, the answers, or the concatenation of both. FAQ retrieval can be broadly divided into 4 categories: lexical, supervised, unsupervised, and conversational.

Lexical To our knowledge, FAQ-Finder [6] was the first to explicitly study the task of FAQ retrieval, it tries to do so by matching user queries to FAQ questions of the Usenet dataset with TF-IDF. FAQ-Finder was later improved by including the similarity to the answer (on top of the similarity to the question) [23]. Another improvement comes from adding a rule-based layer on top of the TF-IDF module [22].

Unsupervised Another approach is to used unsupervised techniques to retrieve the right FAQ pair given a new user query. One possible way is to use Latent Semantic Analysis (LSA) to overcome the lexical mismatch between related queries [11].

Supervised The first supervised methods were developed using tree kernels and SVMs [15]. BERT methods were later developed specifically for the task of FAQ retrieval [20].

Conversational In this paper, we propose a fourth type not yet explored in the literature: conversational. FAQ retrieval can be treated as a special case of conversational modeling: retrieving the answer is similar to retrieving the next utterance in a conversation.

Dual-encoder architectures, pre-trained on response selection, have become increasingly popular in the dialog community due to their simplicity and ease of control [8, 2]. There are two options when it comes to retrieving the next utterance. One can either encode the two sentences separately (dual-encoder) [7], or simultaneously (cross-encoder) [3]. Dual-encoders are faster than cross-encoders as they can cache the answer representations. ConveRT [7] is a dual-encoder pre-trained on a large-scale conversational dataset. Thanks to various design optimizations (such as using single-headed self-attention) ConveRT can vastly reduce the size of the model.

In this work, we choose to focus on ConveRT as it has a low computational cost and does not require a GPU for inference.

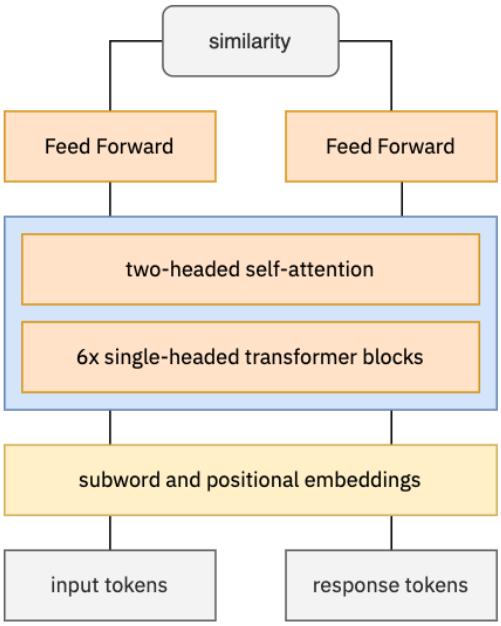


Fig. 1. Illustration of the ConveRT model architecture. The model has three distinct parts. First, the subword and positional embeddings. Second, a shared Transformer block followed by a two-headed self-attention. Third, separate feed-forward networks (3 layers) for the input and responses.

3 ConveRT

In this section, we give a brief overview of the ConveRT (Conversational Representations from Transformers) model [7]. The objective of the model is to generate vector representations for utterances that are as similar as possible (in terms of dot-product) for a given pair. ConveRT takes as input the sequence of tokens of the two utterances. Both sequences are tokenized using the same byte pair encoding vocabulary.

3.1 Architecture

The ConveRT architecture (Fig. 1) is composed of three distinct parts: the embedding layer, the Transformer block and the feedforward layers.

Embedding The first element stores the embeddings for the subwords and position tokens. Embeddings are shared for the input and response representations. Unlike the original Transformer architecture [24], ConveRT uses two positional encoding matrices of different sizes to handle sequences larger than seen during training. We refer the reader to the original paper for a detailed description [7].

Transformer Block The next element is the Transformer block. It closely follows the original Transformer architecture [24] with some notable differences. First, the model uses a single-headed self-attention using a 64-dimensional projection for computing the attention weights. Second, the model applies a two-headed self-attention after the six Transformer layers. The parameters of the Transformer block are fully shared for the input and response sides. ConveRT uses the square-root-of-N reduction [2] to convert the embedding sequences to fixed-dimensional vectors.

Feed Forward The last elements are a series of feed-forward hidden layers with skip connections. The parameters are not shared between the inputs and responses side, as there is a separate feed-forward for the inputs and responses.

3.2 Training Objective

The training objective of ConveRT is to select the right response given a question from a question-answer pair. The relevance of each response to a given input is quantified with a dot-product between the input and response representation. Training proceeds in a batch of K pairs of utterances. The objective is to distinguish between the true relevant responses and irrelevant negative examples (we use other responses from the batch as negative examples). ConveRT uses cross-entropy as the loss function. The model is optimized with Adam [12] and L2 weight decay. The learning rate is warmed up over the first 10,000 steps to a peak value and then linearly decayed.

4 ConveRT for Dutch

In this section, we explain our approach to training a ConveRT model for Dutch. To overcome the limited supply of conversational data available in Dutch, we use a two-stage pre-training: general pre-training on a large open-domain corpus, and conversational pre-training using a smaller conversational dataset from Reddit.

4.1 Data

The original ConveRT model was developed for English using a large-scale conversational dataset from Reddit. We did not have access to such a dataset for Dutch. Instead, we chose to split the problem in two. First, we pre-train the model on a general Dutch corpus. Second, we use a smaller Dutch conversational corpus from Reddit.

General Dataset We consider the same Dutch-language corpora as Bertje [5], a successful Dutch BERT model:

- Books: a collection of contemporary and historical fiction novels

- TwNC [17]: a Multifaceted Dutch News Corpus
- SoNaR-500 [16]: a multi-genre reference corpus
- Web news
- Wikipedia

In total, this is about 12GB of uncompressed text.

To match the setup expected by ConveRT (the tokens of a pair of utterances), we first split each paragraph into sentences. Next, we save pairs of sentences and treat them as pairs of input and response. To avoid small inputs, we filter out pairs with less than 64 characters. After transformation, the general corpus dataset for pre-training has 110M pairs.

Conversational Dataset We also consider a Dutch conversational dataset for which we downloaded comments from around 200 Dutch subreddits. Non-Dutch comments were filtered out. After filtering for the language we arrive at a size of 400K pairs of utterances.

4.2 Pre-training

We followed the training procedure of ConveRT, except for the number of epochs and the batch size. For the general pre-training, we trained the model for 8 epochs. To facilitate the training, we used other examples from the batch as negative examples.

To increase the difficulty of the training, we doubled the batch size at every second epoch. The batch size increased from 128 at the first epoch to 2048 at the last epoch. The larger the batch size, the harder it is for the model as the model has to select the correct response amongst more negative examples.

For the conversational pre-training, we trained for 10 epochs with a fixed batch size of 2048.

model	**split 1**	**split 2**	**split 4**	**split 6**	**split 8**	**split 10**
RASA (baseline)	22%	42%	50%	55%	61%	65%
without pre-training	20%	25%	33%	45%	52%	65%
general pre-training	30%	36%	40%	55%	58%	43%
conversational pre-training	40%	44%	55%	63%	66%	69%
general + conversational pre-training	**46%**	**57%**	**68%**	**69%**	**75%**	**79%**

Table 1. Accuracy on the COVID-19 vaccination FAQ dataset per splits of increasing size. Split one has one training example per answer, while split ten has ten training examples. Pre-training ConveRT on both a general dataset, as well as a conversational dataset provides the best results on this task.

5 Experiments

In this section, we fine-tune our model on a corpus of FAQs related to the COVID-19 vaccine. We then perform an ablation study to analyze which part of the pre-training has the most impact on the downstream performance. To have a better understanding of how our model would perform in the real world, we study its performance as the number of training examples increases.

5.1 Data

We test the performance of our model on a proprietary dataset. The dataset was collected while running a COVID-19 vaccination FAQ bot with Rasa. It consists of 1,200 questions for 76 distinct answers.

5.2 Baseline

As our higher objective is to use this model in a Rasa chatbot, we compare our Dutch ConveRT model to a baseline response retrieval model developed by Rasa.[1] All models are trained using the same number of epochs and dropout probability.

5.3 Low Data Scenario

When starting out, FAQ bots usually have a one-on-one mapping between the number of questions and answers (one question for one answer). As the number of users increases, the number of available questions per answer also increases. To evaluate the generalization capabilities of our model in a low data scenario, we artificially create datasets of increasing sizes, which we call splits. The first split has one training example per answer (the same as when someone starts a new FAQ chatbot), the second split has two training examples per answer, and so on until split ten. We also generate a test set by randomly selecting (and removing from the training set) one training example per answer.

5.4 Results

Results in Table 1 confirm our intuition that the baseline accuracy of the Rasa model radically improves with the number of training examples. In our analysis, the accuracy increases by a factor of 3 from split 1 to split 10. The results also show that a ConveRT model without any pre-training underperforms the baseline, on every split. General pre-training modestly improves the model's performance, but the results are not significantly different from the baseline. Conversational pre-training alone (without any general pre-training) shows a consistent improvement over the baseline. The gain is more visible in the low data regime than in the high data regime. The Dutch ConveRT model reveals its true power when pre-trained on a general corpus and a conversational corpus as it outperforms the baseline by a wide margin on every split.

[1] Rasa does not have a published paper describing their model.

6 Conclusion

We have successfully pre-trained, fine-tuned, and evaluated a Dutch ConveRT model. This model consistently outperforms a baseline response selector from Rasa on a COVID-19 vaccine FAQ dataset.

Conversational datasets for non-English languages are scarce. Our two-phase pre-training procedure bypasses this problem by first pre-training on a general corpus, then pre-training on a smaller conversational corpus.

In future work, we plan on extending the two-stage training to additional languages and additional domains.

7 Acknowledgments

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme. We also thank the reviewers for their helpful comments.

References

1. Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. Rasa: Open source language understanding and dialogue management. *CoRR*, abs/1712.05181, 2017.
2. Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November 2018. Association for Computational Linguistics.
3. Sonam Damani, Kedhar Nath Narahari, Ankush Chatterjee, Manish Gupta, and Puneet Agrawal. Optimized transformer models for faq answering. In Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan, editors, *Advances in Knowledge Discovery and Data Mining*, pages 235–248, Cham, 2020. Springer International Publishing.
4. Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. Bart for knowledge grounded conversations. In *Converse@KDD*, 2020.
5. Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. BERTje: A Dutch BERT Model. arXiv:1912.09582, December 2019.
6. Kristian Hammond, Robin Burke, Charles Martin, and Steven Lytinen. Faq finder: a case-based approach to knowledge navigation. In *Proceedings the 11th Conference on Artificial Intelligence for Applications*, pages 80–86. IEEE, 1995.
7. Matthew Henderson, Iñigo Casanueva, Nikola Mrksic, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulic. Convert: Efficient and accurate conversational representations from transformers. *CoRR*, abs/1911.03688, 2019.
8. Matthew Henderson, Ivan Vulic, Daniela Gerz, Iñigo Casanueva, Pawel Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrksic, and Pei-Hao Su. Training neural response selection for task-oriented dialogue systems. *CoRR*, abs/1906.01543, 2019.

9. Valentin Jijkoun and Maarten de Rijke. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 76–83, 2005.
10. Mladen Karan and Jan Šnajder. Faqir–a frequently asked questions retrieval test collection. In *International Conference on Text, Speech, and Dialogue*, pages 74–81. Springer, 2016.
11. Harksoo Kim and Jungyun Seo. Cluster-based faq retrieval using latent term weights. *IEEE Intelligent Systems*, 23(02):58–65, 2008.
12. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
13. Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*, 2021.
14. Ehsan Lotfi, Maxime De Bruyn, Jeska Buhmann, and Walter Daelemans. Teach me what to say and i will learn what to pick: Unsupervised knowledge selection through response generation with pretrained generative models, 2021.
15. Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. Exploiting syntactic and shallow semantic kernels for question answer classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 776–783, 2007.
16. Nelleke Oostdijk, Martin Reynaert, Véronique Hoste, and Ineke Schuurman. *The Construction of a 500-Million-Word Reference Corpus of Contemporary Written Dutch*, pages 219–247. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
17. Roeland J.F. Ordelman, Franciska M.G. de Jong, Adrianus J. van Hessen, and G.H.W. Hondorp. Twnc: a multifaceted dutch news corpus. *ELRA Newsletter*, 12(3-4), 2007.
18. Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu O Mittal, and Yi Liu. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 464–471, 2007.
19. Anna Rogers, Matt Gardner, and Isabelle Augenstein. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *arXiv preprint arXiv:2107.12708*, 2021.
20. Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. Faq retrieval using query-question similarity and bert-based query-answer relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1113–1116, 2019.
21. Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*, 2021.
22. Eriks Sneiders. Automated faq answering: Continued experience with shallow language understanding. In *Question Answering Systems. Papers from the 1999 AAAI Fall Symposium*, pages 97–107, 1999.
23. Noriko Tomuro and Steven L Lytinen. Retrieval models and q and a learning with faq files. In *New Directions in Question Answering*, pages 183–202, 2004.
24. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

Using Bisimulation Metrics to Analyze and Evaluate Latent State Representations

Nele Albers[1][0000−0002−0502−6176], Miguel Suau[1], and Frans A. Oliehoek[1]

Intelligent Systems, Delft University of Technology, Delft, The Netherlands
{n.albers, m.suaudecastro, f.a.oliehoek}@tudelft.nl

Abstract. Deep Reinforcement Learning (RL) is a promising technique towards constructing intelligent agents, but it is not always easy to understand the learning process and the factors that impact it. To shed some light on this, we analyze the Latent State Representations (LSRs) that deep RL agents learn, and compare them to what such agents should ideally learn. We propose a crisp definition of 'ideal LSR' based on a bisimulation metric, which measures how behaviorally similar states are. The ideal LSR is that in which the distance between two states is proportional to this bisimulation metric. Intuitively, forming such an ideal representation is highly favorable due to its compactness and generalization properties. Here we investigate if this type of representation is also desirable in practice. Our experiments suggest that learning representations that are close to this ideal LSR may improve upon generalization to new irrelevant feature values and modified dynamics. Yet, we show empirically that the extent to which such representations are learned depends on both the network capacity and the state encoding, and that with the current techniques the exact ideal LSR is never formed.

Keywords: Deep Reinforcement Learning · Bisimulation Metrics.

1 Introduction

Recent years have seen a surge of algorithms and architectures for deep Reinforcement Learning (RL), many of which have shown remarkable success for various problems. Yet, little work has attempted to relate the performance of these algorithms and architectures to what the resulting deep RL agents actually learn, and whether this corresponds to what we suppose they should ideally learn. Such a comparison may allow for both an improved understanding of why certain algorithms or network architectures perform better than others and the development of methods that specifically address discrepancies between what is and what should be learned. We thus explore empirically the Latent State Representations (LSRs) a deep RL agent forms of its environment to see whether these match our theoretical expectations.

When we speak of what a deep RL agent learns, we mean the internal representation that a neural network forms of the environment. That is, the activation patterns that arise in each hidden network layer as the result of feeding

(histories of) observations to the network. As the observation space is potentially very large and the capacity of an RL agent is limited, an agent has to learn what to attend to when creating this internal representation. A robot that is trained to fight fires in a residential area, for instance, might learn that certain features such as the house colors do not matter. If so, it will map two observations that differ only in this feature to the same activation pattern. The house color will then no longer influence the action choices, as the agent has learned to ignore it.

Among the desirable properties of such an LSR are that it should make only necessary distinctions between (histories) of observations, allow the agent to learn to act optimally, and enable generalization to new irrelevant feature values and modified dynamics. An LSR that has these properties is one in which the Euclidean distances between states are proportional to a bisimulation metric [6], which measures how "behaviorally different" [7] states are. As such an LSR makes only those distinctions that are needed for the prediction of the next reward and state [12], we call it the *Coarsest Markov State Representation* (CMSR). It is this CMSR that we suppose a deep RL agent should ideally learn. Our main contribution is that we propose a way to measure the degree to which the CMSR is learned, and use this measure to gain insights into the learning process of deep RL agents using Deep Q-Networks (DQNs) [22] as example. Moreover, we show empirically that learning closer to the CMSR may lead to better generalization to new irrelevant feature values and modified dynamics. These evaluations are based on differences in the Markovianity of LSRs that either occur naturally or are obtained via a novel auxiliary loss that pushes a DQN to learn the CMSR.

2 Related Work

Exploring the Learning of Deep RL Agents. Our main goal is to contribute to a better understanding of the learning process of deep RL agents. To this end, we propose using measures based on bisimulation metrics that quantitatively denote how Markov an LSR is. Other research has used saliency maps [13] or t-SNE plots [22][25], the latter of which we also use as supporting evidence. These approaches result in figures that are easy to understand, but they do not produce quantitative measures to effectively summarize the characteristics of an LSR. Instead, to compare state representations, one has to look at multiple images and deduce based on domain knowledge what an agent has learned. An alternative is to plot the test performance [16] or state-action values for certain states [22] during training. Yet, in contrast to our approach, these approaches do not say anything about whether an agent has actually learned or simply memorized [14], the latter of which may hinder generalization. Although offering some improvement, this also holds for measuring out-of-distribution generalization [4][26]. The reason is that such out-of-distribution generalization may be good even if the agent has largely memorized. Lastly, to the best of our knowledge, no prior work has analyzed the learning process by computing how similar to the CMSR an LSR is.

Representation Learning Based on Bisimulation Metrics. To investigate the properties of LSRs that are more similar to the CMSR, we design

an auxiliary loss based on bisimulation metrics. Related work in this regard is presented by [25], who also propose learning LSRs based on bisimulation metrics. Yet, while [25] create an LSR in which distances between states correspond to how behaviorally different they are *under a varying policy*, we take *all actions* into consideration. Thus, an LSR learned by means of the approach of [25] potentially makes fewer distinctions than are needed to predict the reward and next state for all actions. Such an LSR hence generalizes to only a subset of the changes made to the dynamics that still allow for generalization based on the LSR that we propose to learn. In a similar vein to [25], [1] also base their approach on π-bisimulation metrics. Another related work is the one by [11]. Yet, whereas the Euclidean distances in our proposed LSR are *proportional* to the distances assigned by a bisimulation metric, the Euclidean distances between states in the LSR learned by means of the auxiliary loss of [11] provide an *upper bound* to bisimulation metric-based distances. Lastly, [23] employ the more general notion of *MDP homomorphism metrics* for representation learning. MDP homomorphism metrics differ from bisimulation metrics in that actions are also abstracted.

Representation Learning Based on Other Notions. The auxiliary loss we design introduces a bias to the learning. Several other approaches to bias the representation learning of deep RL agents have been proposed. For example, [17] and [8] put forward auxiliary losses based on predicting the next reward or the discount factor. Such methods tend to be successful in practice, but do not have strong theoretical foundations. Other work such as [19] is based on forming a model of the environment as auxiliary task. Yet, this tends to not work well for high-dimensional observations with large amounts of irrelevant information. Furthermore, rather than biasing the learning of deep neural networks by means of auxiliary losses, other work has proposed different models to learn more useful representations such as by incorporating ideas from symbolic reasoning [10]. For instance, [24] constrain neural networks to capture typical characteristics of relational reasoning. Another approach to learning more useful representations is to specifically focus on factors that may hurt generalization. [16], for example, improve generalization by reducing the non-stationarity an agent encounters during training. Moreover, [15] adapt to RL several regularization techniques from the context of classification that are based on injecting noise during training.

3 Background

Markov Decision Process. An infinite-horizon Markov Decision Process (MDP) is a tuple $\langle S, A, P, R, \gamma \rangle$ where S and A describe the space of Markov states and possible actions, respectively, $P : S \times A \to \Pi(S)$ is the transition function such that $P(s'|s, a) \in [0, 1]$ is the probability of arriving in state s' after taking action a in state s, $R : S \times A \to \mathbb{R}$ is the reward function such that $R(s, a)$ is the instant reward for taking action a in state s, and $0 \leq \gamma \leq 1$ is a discount factor. The goal of an agent in an MDP is to learn an optimal policy $\pi^* : S \to \Pi(A)$ that maximizes the expected cumulative (discounted) reward $\mathbb{E}\big[\sum_t^\infty \gamma^t r_t\big]$ for acting in the given environment. The Q-value function $Q^\pi : S \times A \to \mathbb{R}$ describes

the expected cumulative reward for taking action a in state s and executing π thereafter. The expected cumulative reward for taking an action a in a state s and following an optimal policy afterwards is given by $Q^*(s,a)$, where $Q^* = \max_\pi Q^\pi$.

Bisimulation Metrics. Bisimulation metrics [6] are based on the notion of *stochastic bisimulation* [12], which considers states as equivalent if and only if they have the same expected reward and the same transition distribution over all other abstract states for all actions. Such states that are equivalent under the notion of stochastic bisimulation are called *bisimilar*. Bisimulation metrics can be regarded as a quantitative version of stochastic bisimulation in that they assign a distance of zero only to bisimilar states, and that if the parameters of two bisimilar states are altered on a small scale, the metric distance between the two states will stay small. Thus, bisimulation metrics can be seen as a measure of behavioral similarity [7]. Theorem 4.5 in [6] defines one bisimulation metric d_{fix} that considers states as equivalent *if and only if* they are bisimilar. Given $F : M \to M$, where M is the set of all semimetrics on S that assign distances of at most 1, this d_{fix} is defined as the least fixed point of the following equation:

$$F(d)(s,s') = \max_{a \in A} \Big(c_R |R(s,a) - R(s',a)| + c_T T_K(d)\big(P(s,a), P(s',a)\big) \Big). \quad (1)$$

c_R and c_T are two positive one-bounded constants and $T_K(d)$ is the Kantorovich distance. It is d_{fix} that Euclidean distances in the CMSR are proportional to.

4 Markovianity of LSRs During Learning

Here we analyze the LSRs deep RL agents naturally form of their environments and how they compare to what such agents should ideally learn.

4.1 Methodology

Measuring Characteristics of LSRs. We propose using Pearson correlation coefficients[1] to gain insights into the learning process. These correlation coefficients are based on (components of) bisimulation metrics one the one hand, and the Euclidean distances between the activations states are mapped to in a network layer on the other hand. Let z_i, z_j be the activations s_i, s_j are mapped to in a network layer, $d_E(z_i, z_j)$ the Euclidean distance of z_i and z_j, $d_B(s_i, s_j)$ the distance of s_i and s_j for some bisimulation-based measure, and $\overline{d_E}$ and $\overline{d_B}$ averages. Then the Pearson correlation coefficient r_{d_B} is:

$$r_{d_B} = \frac{\sum_{i=0}^{|S|-2} \sum_{j=i+1}^{|S|-1} (d_E(z_i,z_j) - \overline{d_E})(d_B(s_i,s_j) - \overline{d_B})}{\sqrt{\sum_{i=0}^{|S|-2} \sum_{j=i+1}^{|S|-1} (d_E(z_i,z_j) - \overline{d_E})^2} \sqrt{\sum_{i=0}^{|S|-2} \sum_{j=i+1}^{|S|-1} (d_B(s_i,s_j) - \overline{d_B})^2}}. \quad (2)$$

Using measures based on or inspired by bisimulation metrics for d_B leads to the Pearson correlation coefficients that are defined in Table 1. These correlation

[1] The Pearson correlation coefficient measures the linear correlation of two variables.

Table 1. Correlation coefficients (CCs) based on Equation 2 and their interpretations. $r_{d_{fix}}$ and r_{Rew} are based on (components of) bisimulation metrics, and r_{Q^*} replaces the immediate reward in r_{Rew} by Q^*.

CC	d_B	Interpretation
$r_{d_{fix}}$	$d_{fix}(s_i, s_j)$	Similarity of representation to CMSR.
r_{Rew}	$\max_{a \in A} \lvert R(s_i, a) - R(s_j, a) \rvert$	Degree of clustering based on rewards.
r_{Q^*}	$\max_{a \in A} \lvert Q^*(s_i, a) - Q^*(s_j, a) \rvert$	Similarity to Q^*-irrelevance abstraction.

coefficients allow us to analyze the degrees to which the CMSR is learned, states are grouped based on instant rewards, and states are clustered based on Q-values in an LSR. Moreover, we can formally define the CMSR based on $r_{d_{fix}}$, which is obtained by letting d_B in Equation 2 be the bisimulation metric d_{fix}[2].

Definition 1 (Coarsest Markov State Representation (CMSR)). *The CMSR is a representation for which the following holds:*

$$r_{d_{fix}} = 1. \tag{3}$$

Theoretical Properties of the CMSR. We suppose that a deep RL agent should ideally learn the CMSR. This is due to several desirable theoretical properties of this representation. These theoretical properties arise because 1) the CMSR makes the lowest number of distinctions that still enables the prediction of the reward and next state [12], and 2) Euclidean distances between states in the CMSR are proportional to how behaviorally different states are. This leads to the following advantageous characteristics of the CMSR:

- *Feasibility of Learning π^*.* If an agent can predict the next reward and state for each action, an LSR is said to be *Markov* and the agent may find an optimal policy based on (histories of) observations[3] [21]. If, however, the reward and next state cannot be predicted based on the LSR, the agent in the most general case cannot learn an optimal policy.
- *Indifference to Irrelevant Features.* The CMSR does not distinguish observations that refer to the same state in the abstract MDP. That is, the CMSR treats as equivalent two observations that differ only in features that are irrelevant for predicting next states and rewards. This is especially important for domains with high-dimensional observations such as images.
- *Generalization to Modified Dynamics.* If a subset of the features required for predicting the reward and next internal state for an original domain is sufficient for predicting the reward and next internal state after modifying the dynamics, the distinctions the CMSR makes for the original domain

[2] Computed via the MCFZIB solver [9].

[3] While *representing* an optimal policy may require solely a coarser abstraction of the state space, such a representation may not suffice for *learning* an optimal policy [21].

suffice to learn the Q-values of such a modified domain. Moreover, since the Euclidean distance between two states in the CMSR varies smoothly as their parameters are changed, the CMSR is likely to still be useful if small such changes are made. This is important, as dynamics are commonly estimated and domain shifts may arise in problems such as robotics [15].

Q^*-irrelevance Abstraction. We suppose that LSRs should ideally be similar to the CMSR. Yet, the output layer of a DQN is pushed to represent Q-values, which may also cause LSRs to do so. We call an LSR in which the Euclidean distances between activations are proportional to the Euclidean distances between the corresponding Q-values a *Q^*-irrelevance abstraction*. This definition is based on generalizing the levels of state abstraction by [18] to the Euclidean space in which the activations in network layers fall. As non-bisimilar states may have the same Q-values, such an LSR may make fewer distinctions than the CMSR and hence no longer preserve the one-step model. Thus, a Q^*-irrelevance abstraction may not have the theoretical properties of the CMSR. We measure the extent to which a Q^*-irrelevance abstraction is formed via the correlation coefficient r_{Q^*}.

Domain. Our results presented here are based on a modified version of the fully observable Gridworld 3x3 domain [5], but supporting results from Gridworld 5x5, FrozenLake 8x8 from OpenAI Gym and the partially observable Hallway domain are described in [2]. In Gridworld 3x3, the state is a combination of the agent's position on a 3x3 grid and its orientation. Apart from the ground state, the agent's observations in our domain version contain a superfluous feature f_S, which can take 5 possible values sampled uniformly at random. This creates 5 behaviorally identical or bisimilar states out of each ground state. The agent can choose from the deterministic actions $\{forward,\ rotate\}$. The reward is 1 for reaching the goal location in the center of the grid and 0 otherwise.

State Encoding. We one-hot encode the ground states, and use 3 different ways of encoding f_S (Table 2). The encodings vary in the degree to which bisimilar states are encoded similarly, as mirrored by the encoding-based value for $r_{d_{fix}}$ in Table 2. Thus, the encodings have different effects on the initial LSR, which may impact the final LSR and its similarity to the CMSR.

4.2 Analysis of the Learning Process

In the following, we now use our proposed correlation coefficients and t-SNE [20] plots to shed light on the natural learning process of deep RL agents. Fig. 1 shows that the learning process consists of three overlapping learning phases:

1) States are grouped based on multi-step rewards. Since the target network provides the estimates of the Q-values of next states during training, it is not surprising that the activations of states with the same $n+1$-step rewards tend to be grouped together, where n is the number of times the target network has been updated. Fig. 1-1 shows the hidden activation patterns right after the DQN has been initialized[4]. At this point, any clustering is incidental in that it

[4] Since the encoding of f_S is lower-dimensional than the one of the ground state, the t-SNE plot shows one cluster for each value for f_S rather than for each ground state.

Table 2. State encodings and their definition of the superfluous feature f_S. We also show the value for $r_{d_{fix}}$ based on the encoded states.

Encoding	f_S	$r_{d_{fix}}$
Norm (N)	$f_S \in \{0, 0.25, 0.5, 0.75, 1\}$	0.251
One-hot (OH)	$f_S \in \{[1,0,0,0,0], [0,1,0,0,0], [0,0,1,0,0], [0,0,0,1,0], [0,0,0,0,1]\}$	0.087
Original (O)	$f_S \in \{0, 1, 2, 3, 4\}$	0.015

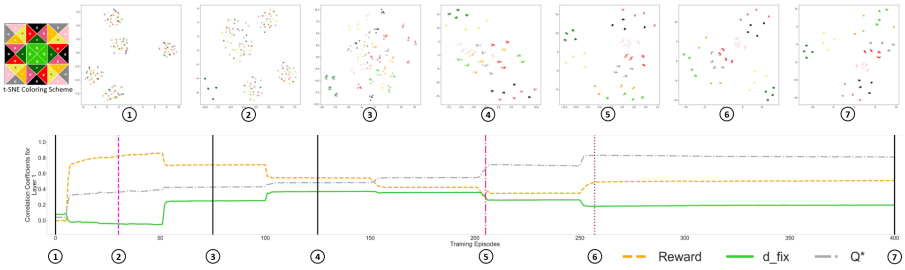
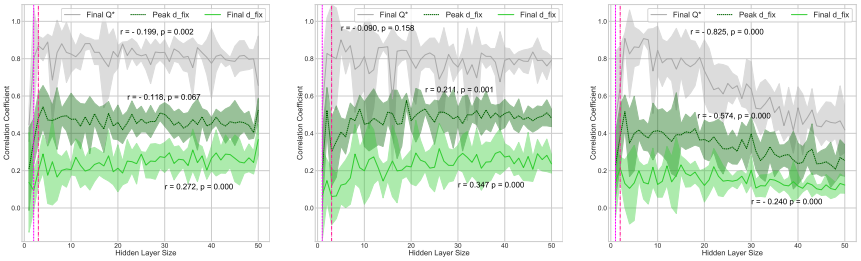


Fig. 1. r_{Rew}, $r_{d_{fix}}$, r_{Q^*} and t-SNE plots of the activations observations are mapped to during training for the LSR of a 2-layer DQN for the OH-encoding. The hidden layer size is 50 and the target network is updated every 50 episodes. All observations differing solely in f_S are drawn in the same color in the t-SNE plots and the coloring scheme for the ground states is shown on the left. Bisimilar ground states are shown in the same color. The vertical lines mark the episodes for which we show t-SNE plots. The 3 non-black lines thereby indicate 1) the first time the agent reaches the goal in each of 100 test episodes, 2) the first time the agent has learned π^* and 3) convergence to π^*.

depends on the state encoding[5] and network initialization. In Fig. 1-2, we see that the DQN has formed a separate cluster for those states that have an immediate reward of 1 (dark green). The target network has not yet been updated, so all other states, which have an immediate reward of 0, should not yet fall into separate clusters. Also note that the yellow curve (r_{Rew}) is now at its maximum. This is expected, because r_{Rew} measures the degree of similarity between the current LSR and a representation that clusters states together if and only if they have the same immediate reward. After the target network has been updated once, a new separate cluster is formed for those states that have a non-zero two-step reward (Fig. 1-3, dark pink). This is accompanied by a drop in r_{Rew}, as states are now no longer distinguished solely based on their immediate rewards.

2) The LSR becomes more similar to the CMSR. This pattern is mirrored by the increase in the green curve ($r_{d_{fix}}$) at the beginning of training. However, the exact CMSR is not learned, as $r_{d_{fix}}$ is never equal to 1.

[5] The impact of the state encoding is discussed in the next section.



(a) OH-encoding. (b) N-encoding. (c) O-encoding.

Fig. 2. Mean peak and final $r_{d_{fix}}$ and final r_{Q^*} with 95%-confidence intervals for the LSRs of 2-layer DQNs for different state encodings and hidden layer sizes. The vertical lines indicate the smallest hidden layer sizes for which 1) the agent always arrives at the goal in 100 test episodes and 2) the DQN converges to π^* at least 1 out of 5 times. Each curve is labeled with the Pearson correlation of the respective correlation coefficient and the hidden layer sizes that are large enough for the DQN to learn π^* at least 1 out of 5 times.

3) States are increasingly clustered based on Q-values, as visualized by the step-wise increase in the gray curve (r_{Q^*}), after an initial plateau. Ultimately, r_{Q^*} reaches a value near 1 when the DQN converges to π^*. At the same time, $r_{d_{fix}}$ decreases for this domain as the inter-cluster distances become more and more different from those of the CMSR[6]. This is shown near episode 200, where $r_{d_{fix}}$ begins to decrease when r_{Q^*} strongly increases again. The final LSR is thus less similar to the CSMR for this domain than during the second phase.

This analysis suggests that while a DQN does naturally form the CMSR to some degree, the exact CMSR is not learned. Instead, states are at some point clustered based on Q-values rather than bisimilarity, which may cause the LSR to become less similar to the CMSR. Given the useful theoretical properties of the CMSR, the latter might have negative consequences for a network's generalization ability. We examine this impact on the generalization performance in Section 5.

4.3 Factors Impacting the Learning Process

When training a DQN, one has to make a plethora of choices such as for the network architecture and the state encoding. Commonly, we make such choices primarily based on average returns. However, the decisions we make might also impact the LSRs that are formed. We therefore analyzed how different factors impact the learning process described above. We find that the extent to which LSRs become similar to the CMSR *during* and still are *at the end of* training depends on the network capacity and state encoding. This is discussed below.

Network Capacity. The dark green curve (*peak* $r_{d_{fix}}$) in Fig. 2(a) shows that the LSR becomes most similar to the CMSR *during* training for hidden layer sizes just to the right of the second vertical line. These hidden layer sizes

[6] The decrease in $r_{d_{fix}}$ is related to the network capacity, discussed in the next section.

are necessary for the DQN to be able to converge to π^*. For larger hidden layers, the LSR becomes progressively less similar to the CMSR during training. This is captured by the value of -0.118 for the Pearson correlation coefficient between *peak* $r_{d_{fix}}$ and sufficiently large hidden layer sizes (Fig. 2(a)). The reason for this pattern is that larger hidden layers make a network more flexible, and thus allow the network to converge to the true Q-values even if less similar to the CMSR is learned in the hidden layer during training. Such large networks hence learn the Q-values without grouping behaviorally equivalent observations together.

The LSR *at the end of* training, however, is more similar to the CMSR for larger hidden layers. This is indicated by the bright green curve (*final* $r_{d_{fix}}$) and the corresponding Pearson correlation coefficient of 0.272 with respect to sufficiently large hidden layer sizes in Fig. 2(a). The reason is that DQNs with smaller hidden layers eventually need to largely cluster states based on Q-values in their hidden layers due to their lower flexibility. Otherwise, their output layers cannot represent the true Q-values. Thus, while DQNs with smaller hidden layers *initially* learn closer to the CMSR, their LSR is *ultimately* further abstracted towards a Q^*-irrelevance abstraction. The latter is supported by the observation that the final values for r_{Q^*} (gray curve) are higher for smaller hidden layers, which is captured by the Pearson correlation coefficient of -0.199 between the final values for r_{Q^*} and sufficiently large hidden layer sizes in Fig. 2(a).

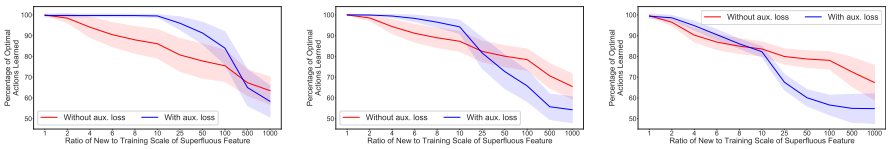
State Encoding. The CMSR is formed to a lesser degree during learning if it is more difficult and less necessary to be learned. Based on the three dark green curves (*peak* $r_{d_{fix}}$) in Fig. 2, we can see that the LSRs become most similar to the CMSR *during* learning for large hidden layers for the N-encoding and least similar for the O-encoding. The reason for this pattern is that bisimilar states have the most similar encodings in the N- and the least similar ones in the O-encoding (see $r_{d_{fix}}$ in Table 2). Hence, for the latter encoding it is most difficult to group bisimilar states together in the LSR. Thus, as the network capacity increases and it therefore becomes less necessary to learn the CMSR, the CMSR is progressively less formed *during* learning for state encodings that make it more difficult to do so. This also impacts the LSRs present *at the end of* training, as mirrored by the three bright green curves (*final* $r_{d_{fix}}$) in Fig. 2.

Given that both the network capacity and the state encoding impact the degree to which the CMSR is formed, it is important to make a considerate choice of the network architecture and state encoding if learning the CMSR is desired.

5 Practical Usefulness of the CMSR

While the *theoretical* advantages are apparent, we will now investigate whether striving to learn the CMSR is also useful *in practice*. To obtain LSRs that are very similar to the CMSR, we introduce a bisimulation-based auxiliary loss that pushes a network to form the CMSR as LSR.

Bisimulation-based Auxiliary Loss. We calculate $d_{fix}(s_i, s_j)$ for all $s_i, s_j \in S, i \neq j$. During training, we then compute an auxiliary loss based on the premise that we want the Euclidean distances between the activations of states to be



(a) Hidden layer size of 10. (b) Hidden layer size of 20. (c) Hidden layer size of 65.

Fig. 3. Average percentage of optimal actions learned by 2-layer DQNs with different hidden layer sizes for the O-encoding, trained with and without the auxiliary loss. Optimal actions returned by the DQN for each non-terminal ground state are measured for $1,000$ values for f_S sampled uniformly at random from an interval that is i times as large as the one used during training. The value i is shown on the x-axis. 95%-confidence intervals based on 10 repetitions are shown.

proportional to their distances assigned by the bisimulation metric d_{fix}. In other words, we want that $d_E(z_i, z_j)$ is equal to $d^*_E(z_i, z_j) = d^{max}_E \times d_{fix}(s_i, s_j)$, where d^{max}_E is a hyperparameter for how far apart the activations of non-bisimilar states should be. We thus compute a target activation z^*_i for all $s_i \in S$:

$$z^*_i = z_i + \frac{1}{2} \times \sum_{j \neq i} \big(d^*_E(z_i, z_j) - d_E(z_i, z_j)\big) \frac{z_i - z_j}{||z_i - z_j||}, \tag{4}$$

where $||z_i - z_j||$ is the length of the vector $z_i - z_j$. Note that the unit-length vector $\frac{z_i - z_j}{||z_i - z_j||}$ between z_i and z_j is multiplied by half of the amount by which $d_E(z_i, z_j)$ should change. The idea behind this is that if z_i and z_j should be pulled apart or closer together, both are moved by half the total amount in the respective direction. Based on this, we minimize the MSE between z_i and z^*_i for all $s_i \in S$. We found this approach to work better than directly minimizing the MSE between d_E and d^*_E.

5.1 Generalization to New Irrelevant Feature Values

The first type of generalization we consider is the one to new values of irrelevant[7] features. We train 2-layer DQNs for Gridworld 3x3 with and without the auxiliary loss. At test time, we sample $1,000$ values for the superfluous feature f_S randomly from an interval that is i times as large as the one used during training, where $i \in \{1, 2, 4, 6, 8, 10, 25, 50, 100, 500, 1000\}$. For each sampled value for f_S, we compute the optimal action returned by the trained DQN and compare it to π^*.

Fig. 3 reveals that if the auxiliary loss is used, the generalization to new values for f_S tends to be better than if no auxiliary loss is used. This makes sense, as using the auxiliary loss causes the LSR to ignore f_S to a larger extent (Fig. 4). However, Fig. 3 shows that there are two exceptions to the observation that introducing the auxiliary loss improves upon the generalization. These are 1) the generalization to very large intervals and 2) DQNs with large hidden layers:

[7] Irrelevant features are not required for predicting the next reward and internal state.

Very Large Intervals. Generalization to values for f_S sampled from very large intervals tends to be better if LSRs that are not entirely indifferent to f_S are closer to a Q^*-irrelevance abstraction. Notice that while introducing the auxiliary loss leads to improved generalization for small and moderately sized intervals, it deteriorates the generalization for very large intervals. This can be explained by the fact that even though the LSRs learn to ignore f_S to a larger extent when we apply the auxiliary loss, they do not do so entirely. At the same time, the Euclidean distances between the activations of states with *different* optimal actions are on average more similar to those between the activations of states with the *same* optimal actions in the CMSR than in a Q^*-irrelevance abstraction for this domain[8]. Hence, that for very different values for f_S an observation is mapped to a latent representation that causes the DQN to return a sub-optimal action is less likely if the DQN learns closer to a Q^*-irrelevance abstraction. Yet, this only holds because the DQNs do not learn the *precise* CMSR.

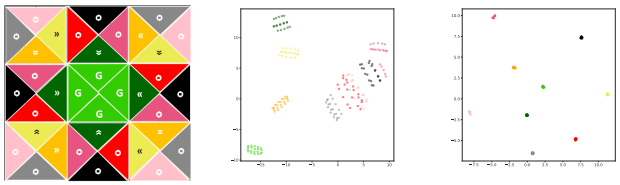
DQNs with Large Hidden Layers. One would expect DQNs with varying hidden layer sizes to generalize similarly well if the LSRs are very close to the CMSR. Yet, using the auxiliary loss tends to lead to worse generalization to large intervals for large hidden layers (Fig. 3(c)) than for smaller ones (Fig. 3(a) and 3(b)). The reason is that the LSRs of DQNs with large hidden layers become less similar to the CMSR again towards the end of training for our settings for the auxiliary loss. More precisely, we decay the weight of the auxiliary loss during training and continue to train even after the weight has become 0. This continued training after the auxiliary loss is no longer applied causes the LSRs of larger DQNs to increasingly distinguish observations based on f_S again and hence to generalize worse to large intervals. Thus, for large DQNs to have an LSR that is very similar to the CMSR by the end of training, it is not sufficient to apply the auxiliary loss only until close to the CMSR is formed. Instead, the auxiliary loss needs to be applied longer, if not during the entire training.

Worse generalization hence only arises when the exact CMSR is not formed. Moreover, even then it only occurs when either extremely different values for f_S are sampled or the auxiliary loss is stopped too soon for very large DQNs.

5.2 Generalization to Modified Dynamics

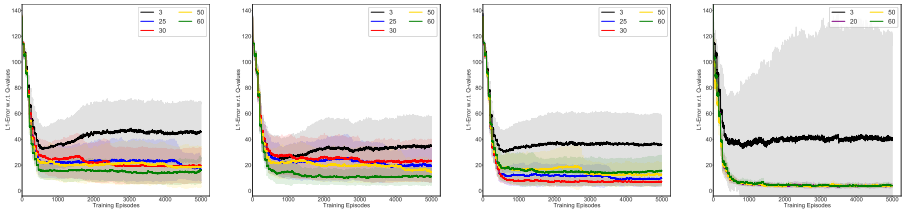
Here we now explore a second type of generalization, namely the one to modifications of the dynamics that do not make formerly irrelevant features relevant. 2-layer DQNs with hidden layer sizes between 3 and 60 are trained each 10 times on Gridworld 3x3, and subsequently retrained after modifying the transition function. We reset the output-layer representation before and hold the LSR fixed during retraining. Based on Fig. 5, we find that the following three factors impact the generalization to the modified domain:

[8] Non-terminal ground states have mean Euclidean distances of 0.175 and 0.333 to other non-terminal ground states with the same and different optimal actions, respectively, in a Q^*-irrelevance abstraction for Gridworld 3x3. In the CMSR, however, the mean Euclidean distances to non-terminal ground states with the same and different optimal actions are 0.141 and 0.144, respectively, if $d_E^{max} = 1$.



(a) Color scheme. (b) $r_{d_{fix}} = 0.11$. (c) $r_{d_{fix}} = 0.98$.

Fig. 4. t-SNE plots and $r_{d_{fix}}$ of the LSRs at the end of training b) without and c) with the auxiliary loss for a 2-layer DQN with a hidden layer size of 10 for the O-encoding. Activation patterns of bisimilar observations have the same color.



(a) N - Without. (b) OH - Without. (c) O - Without. (d) O - With.

Fig. 5. Mean L_1-error with respect to the Q-values during retraining of 2-layer DQNs with varying hidden layer sizes on the modified domain for the Norm (N), One-Hot (OH) and Original (O) state encodings. The hidden-layer weights are initialized to those of DQNs trained on Gridworld 3x3 either *with* or *without* the auxiliary loss and are not updated during retraining. The output-layer weights are newly initialized before retraining. Values are based on 10 repetitions and 95%-confidence intervals are shown.

Similarity to Q^*-irrelevance Abstraction. The generalization is better when the LSR is less similar to a Q^*-irrelevance abstraction for the original domain. Recall that DQNs with larger hidden layers learn LSRs that are less similar to a Q^*-irrelevance abstraction (gray curves in Fig. 2). This explains why DQNs with larger hidden layers tend to generalize best for the Norm (N) and One-Hot (OH) encodings. Moreover, the created LSRs for large hidden layers are closer to a Q^*-irrelevance abstraction for the N- and OH- than for the Original (O)-encoding (gray curves in Fig. 2), which is why the former lead to higher L_1-errors on the modified domain.

Similarity to CMSR. Lower L_1-errors are achieved when the LSR is closer to the CMSR. Moderately sized hidden layers are more similar to the CMSR for the O-encoding than even larger hidden layers (bright green curve in Fig. 2(c)), which is why the former lead to better generalization. Note that this occurs despite the higher flexibility of larger networks. For the OH- and N-encodings, the largest tested hidden layer sizes do not yet cause the final LSR to be less similar to the CMSR (Fig. 2(a) and 2(b)). Thus, DQNs with moderately sized hidden layers do not outperform DQNs with larger ones for those two encodings.

Network Capacity. DQNs with larger hidden layers are less dependent on the LSR when it comes to learning the new Q-values due to their higher capacity. This adds to the fact that larger DQNs generalize best for the N- and OH-encodings. Note also that due to their lower flexibility, DQNs with very small hidden layer sizes need to learn an LSR that is more similar to a Q^*-irrelevance abstraction for the new domain to be able to learn the new Q-values. This is not possible if the LSR is fixed during retraining.

Thus, the naturally occurring differences in Markovianity between LSRs show that learning an LSR that is more similar to the CMSR tends to aid generalization, especially for moderately sized hidden layers. Furthermore, adding an auxiliary loss to the training that pushes a DQN to learn closer to the CMSR in its hidden layer leads to better generalization for all networks except those with very small hidden layers (Fig. 5(d)). The latter occurs because due to their lower flexibility, very small networks need to learn closer to a Q^*-irrelevance abstraction in their hidden layers to be able to learn the new Q-values in their output layers.

6 Conclusions

We analyzed the LSRs deep RL agents form of their environments to gain a better understanding of the learning process and the factors that impact it. Thereby, we suppose that due to its theoretical and especially generalization properties, an agent should ideally learn the CMSR. In the CMSR, distances between states are proportional to how behaviorally different the states are. We find that while LSRs tend to become more similar to the CMSR at the start of training, states are ultimately clustered based on Q-values rather than behavioral similarity. This may cause the LSRs to become less similar to the CMSR again. Moreover, the *precise* CMSR is not learned in any of our experiments. Our standard network architectures and optimization algorithms thus do not lead to ideal LSRs. While our analysis in this paper is based on Gridworld 3x3, we obtained comparable results for the learning process on Gridworld 5x5, FrozenLake 8x8 from OpenAI Gym and the partially observable Hallway domain in [2].

Our analysis of the factors impacting the learning process further reveals that both the state encoding and the network capacity impact the degree to which the CMSR is formed *during* and is still present *at the end of* training. For large hidden layer sizes, for example, networks learn the CMSR to a much lesser extent during training. The reason is that due to their higher flexibility, such networks can learn the Q-values without grouping behaviorally equivalent observations together. Notably, the CMSR is even less learned by such large networks if it is also rather difficult to form the CMSR due to the state encoding. It is thus crucial to carefully choose both network architecture and state encoding if learning closer to the CMSR is desired. Future work should explore the generalization of these findings to environments with more complex observations. For such environments, our proposed correlation coefficients can be made more scalable by approximately computing the bisimulation metric based on the algorithm by [3].

Our claim that deep RL agents should ideally learn the CMSR is supported by our empirical findings. That is, we find that learning closer to the CMSR may improve upon generalization to new irrelevant feature values and modified dynamics. Our results thus show that learning good LSRs is crucial. Rather than selecting architectures and optimization algorithms primarily based on average returns, we should hence strive to make a more informed decision based on the LSRs that are formed. To this end, we need to also report the quality of the LSRs learned in our experiments via measures such as the ones we propose. Moreover, as our current architectures and algorithms do not form ideal LSRs, it is important that we as a community strive to develop scalable methods that address the discrepancies between what is and what should be learned. The auxiliary loss we designed provides a starting point, but has to be made more scalable to be useful in practice. For example, the expensive exact computation of the bisimulation metric could be replaced by an approximation that is incorporated into training in a vein similar to the approach by [3].

Acknowledgments. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824 —INFLUENCE).

References

1. Agarwal, R., Machado, M.C., Castro, P.S., Bellemare, M.G.: Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In: International Conference on Learning Representations, ICLR (2021)
2. Albers, N.: Learning what to attend to: Using bisimulation metrics to explore and improve upon what a deep reinforcement learning agent learns. Master thesis (2020)
3. Castro, P.S.: Scalable methods for computing state similarity in deterministic Markov decision processes. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10069–10076 (Apr 2020)
4. Cobbe, K., Klimov, O., Hesse, C., Kim, T., Schulman, J.: Quantifying generalization in reinforcement learning. In: Proceedings of the 36th International Conference on Machine Learning, ICML. Proceedings of Machine Learning Research, vol. 97, pp. 1282–1289. PMLR (2019)
5. Ferns, N., Castro, P.S., Precup, D., Panangaden, P.: Methods for computing state similarity in Markov decision processes. In: Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence. pp. 174–181. AUAI Press (2006)
6. Ferns, N., Panangaden, P., Precup, D.: Metrics for finite Markov decision processes. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence. pp. 162–169. AUAI Press (2004)
7. Ferns, N., Precup, D.: Bisimulation metrics are optimal value functions. In: Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence. pp. 210–219. AUAI Press (2014)
8. François-Lavet, V., Bengio, Y., Precup, D., Pineau, J.: Combined reinforcement learning via abstract representations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3582–3589 (2019)

9. Frangioni, A., Manca, A.: A computational study of cost reoptimization for min-cost flow problems. INFORMS Journal on Computing **18**(1), 61–70 (2006). https://doi.org/10.1287/ijoc.1040.0081
10. Garcez, A.d., Besold, T.R., De Raedt, L., Földiak, P., Hitzler, P., Icard, T., Kühnberger, K.U., Lamb, L.C., Miikkulainen, R., Silver, D.L.: Neural-symbolic learning and reasoning: Contributions and challenges. In: 2015 AAAI Spring Symposium Series (2015)
11. Gelada, C., Kumar, S., Buckman, J., Nachum, O., Bellemare, M.G.: DeepMDP: Learning continuous latent space models for representation learning. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 2170–2179. PMLR (2019)
12. Givan, R., Dean, T., Greig, M.: Equivalence notions and model minimization in Markov decision processes. Artificial Intelligence **147**(1-2), 163–223 (2003). https://doi.org/10.1016/S0004-3702(02)00376-4
13. Greydanus, S., Koul, A., Dodge, J., Fern, A.: Visualizing and understanding Atari agents. In: Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1792–1801. PMLR (2018)
14. Hausknecht, M., Stone, P.: The impact of determinism on learning atari 2600 games. In: AAAI Workshop on Learning for General Competency in Video Games (2015)
15. Igl, M., Ciosek, K., Li, Y., Tschiatschek, S., Zhang, C., Devlin, S., Hofmann, K.: Generalization in reinforcement learning with selective noise injection and information bottleneck. In: Advances in Neural Information Processing Systems 32. pp. 13956–13968 (2019)
16. Igl, M., Farquhar, G., Luketina, J., Boehmer, W., Whiteson, S.: The impact of non-stationarity on generalisation in deep reinforcement learning. arXiv preprint arXiv:2006.05826 (2020)
17. Jaderberg, M., Mnih, V., Czarnecki, W.M., Schaul, T., Leibo, J.Z., Silver, D., Kavukcuoglu, K.: Reinforcement learning with unsupervised auxiliary tasks. In: 5th International Conference on Learning Representations, ICLR (2017)
18. Li, L., Walsh, T.J., Littman, M.L.: Towards a unified theory of state abstraction for MDPs. In: International Symposium on Artificial Intelligence and Mathematics, ISAIM (2006)
19. Li, X., Li, L., Gao, J., He, X., Chen, J., Deng, L., He, J.: Recurrent reinforcement learning: A hybrid approach. arXiv preprint arXiv:1509.03044 (2015)
20. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
21. McCallum, A.K.: Reinforcement Learning with Selective Perception and Hidden State. Ph.D. thesis (1996)
22. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
23. van der Pol, E., Kipf, T., Oliehoek, F.A., Welling, M.: Plannable approximations to MDP homomorphisms: Equivariance under actions. In: AAMAS (2020)
24. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. In: Advances in Neural Information Processing Systems 30. pp. 4967–4976 (2017)
25. Zhang, A., McAllister, R., Calandra, R., Gal, Y., Levine, S.: Learning invariant representations for reinforcement learning without reconstruction. arXiv preprint arXiv:2006.10742 (2020)
26. Zhang, C., Vinyals, O., Munos, R., Bengio, S.: A study on overfitting in deep reinforcement learning. arXiv preprint arXiv:1804.06893 (2018)

Algorithmic Trading in Experimental Markets with Human Traders: A Literature Survey

Te Bao[1[0000-0001-6397-9365]] , Elizaveta Nekrasova[2[0000-0003-1456-1725]], Tibor Neugebauer[2[0000-0002-1183-7979]] and Yohanes E. Riyanto[1[0000-0001-7343-4043]]

[1]Nanyang Technological University, Singapore
[2] University of Luxembourg, Luxembourg
elizaveta.nekrasova@uni.lu

Abstract. This paper surveys the nascent experimental research on the interaction between human and algorithmic traders in experimental markets. We first discuss studies in which algorithmic traders are in the researcher's hands. Specifically, the researcher assigns computer agents as traders in the market. We then discuss the studies in which the researchers allow human traders to decide whether to employ algorithms for trading. The paper introduces the types and performances of algorithmic traders that interact with human subjects in the laboratory, including zero-intelligent traders, arbitragers, fundamentalists, adaptive algorithms, and manipulators. The potential impact of interactions with algorithms on the investor's psychology is also discussed.

Keywords: Experimental Asset Market, Algorithmic Trading, High-Frequency Trading, Human-Agent Experiments, Survey

1 Introduction

The financial world has witnessed a skyrocketing volume of algorithm trading since the beginning of the 2000s. Today most transactions in financial markets are executed by automated trading systems. According to Treleaven et al. (2013), algorithm trading already accounted for more than 70% of US stocks' trading volume in 2011. Flash crashes, which until May 6, 2010, were unprecedented phenomena of extreme short-term volatility triggered by high-frequency trading (Kirilenko et al. 2017), prominently demonstrate that algorithms have radically changed the financial market environment. It seems fair to say that today, without understanding the impact of algorithmic trading, a thorough understanding of market behavior would become almost impossible.

To understand the behavior, we need answers to many questions. For example, what are the impacts of trading algorithms on market quality, the return to the usage of such algorithms, and how will human traders' returns be affected? How do humans, especially individual investors, respond to trading in markets dominated by algorithms; does interaction with algorithms also generate emotional or psychological responses by human traders that could create more price fluctuations due to market sentiment? Can

financial advice be rationalized with algorithms? What type or features of algorithms are either helpful or rather harmful to investors; what kind of regulation is sensible? We contribute to this research by reviewing the answers given to some of these questions by the experimental literature focusing on the interaction between algorithms and humans in laboratory markets.

There have been some extensive related literature reviews on the real-world financial marketplace. Kirilenko and Lo (2013) provide an initial review of algorithmic trading in the real-world financial marketplace. The authors acknowledge types of automated trading, including passive strategies like market-making, arbitrage trading, and more aggressive high-frequency trading. They recount important historical events in the age of machine trading, including the 2010 flash crash and other cases of high-frequency trading manipulation such as spoofing. Finally, they reflect on potential regulatory measures, particularly in view of the presence of high-frequency trading algorithms, including speed bumps and Tobin taxes. Goldstein et al. (2014) provide a literature survey on algorithmic trading, including theory and studies based on real-world data. Miller and Shorter (2016) survey recent developments in high-frequency strategies, focusing on recent efforts in regulatory measures. Beckhardt et al. (2016) provide a broad survey on high- frequency trading strategies, including simulation analyses of profitability. We refer the interested reader to the surveys mentioned above as these issues go beyond the scope of the current review, which is limited to controlled laboratory studies.

A related area of interest is agent-based modeling, where algorithms interact with algorithms. Duffy (2006) provides an excellent survey. He summarizes the literature on zero-intelligence agents, learning, and evolutionary algorithm models of agent behavior. Duffy also reviews the literature that compares human laboratory results with simulation results. Brewer (2008) and De Luca et al. (2011) also review zero-intelligence agents and their extensions. In the following section, we review some relevant algorithms for the interaction with humans discussed in that literature.

Closer to us in terms of coverage, March (2019) provides a broad survey on the interaction of computer players with human subjects, including experiments on strategic reasoning, social dilemmas, markets, auctions, bargaining and negotiation, and other topics. Naturally, this literature survey overlaps with March's survey, notably with his section on market experiments. Nevertheless, our focused approach allows a more detailed report of the studies and also includes unpublished work.

This paper provides an overview of the experimental literature on algorithmic trading in experimental financial markets, focusing on human-robot interaction. The reported research is interdisciplinary. The interaction of man and machine is of general interest to the behavioral sciences and the computer sciences. The findings of this research can have implications for regulation. That said, the laboratory research that we report here is nascent. Based on the literature survey, we propose, without the ambition of being

conclusive, some interesting questions for future research in this area and possible policy implications.

The remainder of the paper is organized as follows. Section 2 reviews the literature on experimenter-induced algorithms that concentrate on the performance of algorithms versus human traders, studies that look at market quality, the behavioral effects of algorithm speed, manipulation, and arbitraging activity, and the question of subjects' aversion to interacting with algorithms. In section 3, we review studies in which the experimenter puts algorithms in the hands of human subjects. In section 4, we finally conclude and discuss future directions.

2 Experimenter induced algorithms

In this section, we review the literature on experiments involving subjects playing the role of traders competing against algorithms programmed by researchers. Loosely following a historical perspective, we begin by briefly reviewing the literature in which the efficiency of markets populated by algorithms is compared to experimental markets with only human subjects before we look at the performance in hybrid markets. We dedicate a sub-chapter to discuss the effects of the algorithm's reaction time, another to discuss arbitrage in multiple markets, and yet another to cover the manipulation with algorithms. Finally, we turn to how the announcement of possible market participation of an algorithm can impact human subject behavior.

2.1 Comparison of algorithms in simulations with human traders in experimental markets

The early experimental studies involved no interaction between algorithms and human subjects. These studies compare the outcomes in experiments with human subjects to the ones of interacting algorithms in the continuous double auction (hereafter CDA) markets.

The documented academic research on algorithms in asset markets seemingly started in the late 1980s. Shyam Sunder (2003, p. 10f) recounts his approach: the press blamed the stock market crash of 1987 on algorithmic trading. Skeptical of this claim, Sunder designed and taught a course at Carnegie Mellon University on algorithmic trading to learn about the structure of trading strategies and the behavior of the CDA market. Being challenged by the students in the course, he and Gode programmed a random algorithm -later labeled 'zero intelligence or ZI traders (Gode and Sunder 1993), which adheres to a budget constraint. The chosen CDA market environment was Smith (1962), with induced values and costs. ZI traders provide liquidity to the market by repeatedly submitting orders; ZI buyers submit bids between 0 and the experimenter induced value; ZI sellers submit offers between the upper bound of the cost distribution and the induced cost level. Since traders have zero intelligence, they do not profit-maximize, remember nor learn. In Gode and Sunder (1993), a ZI transaction occurs whenever the

best bid exceeds the best offer (each for one unit), the transaction price being equal to the earlier submitted one of the two.

The result was that ZI agents achieved an allocative efficiency of 99% across different sessions, comparable to the one found in data from experiments with human subjects. Gode and Sunder (1993, p. 134) conclude that "the high allocative efficiency of double auctions is [caused by] market discipline imposed on traders" and not by profit maximization, learning, and intelligence. This literature on zero-intelligence traders (summarized in Duffy 2006; De Luca et al., 2011) provides a very important micro-foundation of the general equilibrium theory by showing that market efficiency does not rely on perfect individual rationality and utility maximization behavior. Nonetheless, the trajectories of equilibrium market prices with human subjects are relatively flat, whereas the ZI agents produce continued volatility around the equilibrium price. Duffy and Ünver (2006) report related similarities of price efficiency patterns between ZI agents and human subjects in the Smith et al. (1988) CDA market with a multi-period lived asset,[1] which frequently generates bubbles and crashes in laboratory studies.

Arifovic (1996) finds in an experimental macroeconomic setting that the market price behavior of human experimental subjects shares similarities to that of a genetic algorithm.[2] The genetic algorithm selects a decision rule defined by a binary string (length 30) and is updated using three genetic operations to produce offspring; reproduction, crossover, and mutation.[3]

[1] The design of Smith et al. (1988) is described as follows: Nine subjects, initially endowed with cash and assets, can buy or sell assets between each other during 15 periods in a CDA market. No margin purchases and no short sales are permitted. Assets and cash carry over between periods. At the end of the period, a dividend is paid to the asset holders, which takes one of four values in cash units, {0, 8, 28, 60}, and is independently and identically drawn in each period. At the end of the last period, the assets are redeemed at 0 cash units. Hence, the fundamental dividend value is constantly declining across periods.

[2] Arifovic studies exchange rate behavior in an overlapping generations model with fiat money. Endowed with units of the consumption good in two periods, the decision makers decide on their consumption when young and their savings in two currencies, which both allow the purchase of the consumption good when old. Intertemporal consumption is valued with an utility function, which translates to a fitness value in the genetic algorithm.

[3] Selection involves the random mating of two parental decision rules. The probability of selection of each parent decision rule depends on its fitness value, which is the ex-post value of the utility function. Reproduction implies an identical copy of the binary strings of each parent to begin with. Crossover is the exchange of parts of the initial strings. Mutation is a random change from 0 to 1 or 1 to 0 of a position within a string. The initial two generations in the genetic algorithm are randomly determined decision

Rust et al. (1994) report on the Santa Fe Institute double auction tournament -SFDAT in 1990/91. For the SFDAT, 30 colleagues submitted profit-maximizing algorithms, including quite complex ones, to trade with another in the Smith (1962) CDA market. To their surprise, the tournament winner involved relatively simple liquidity absorbing (profit-making) strategy -later labeled Kaplan's Sniping Agent. The Sniper seller (buyer) sends a limit order to sell (buy) at the market best bid (offer) if at least one of three conditions is met; the best bid (offer) is at least as good or better as the high (low) transaction price of the previous period; the offer-bid spread is small (≤ consts) while the expected profit is more than a minimum profit factor (≥ constP); few instances of time left until the closing of the market period (≤ constt). Later simulation studies highlighted that this Kaplan's sniper could only be profitable if few agents apply it, as it is not the best response against itself.[4]

2.2 Performance of algorithmic and human traders in hybrid experimental-markets

Das et al. (2001) study how agent-human interaction influences human traders' market outcome and trading performance in an experimental asset market setting within a CDA environment with induced values (Smith 1962). In each of their experimental markets, there are 6 human traders and 6 algorithmic traders. The algorithmic trader may adopt two types of adaptive trading strategies: (1) the "zero intelligence plus (ZIP)" algorithm (Cliff 1997) provides liquidity to the market, similarly to ZI. Still, its orders involve a private profit margin updated over time if a limit order fails to transact or transacts immediately. When a trade takes place, all agents adjust their bids towards the transaction price. If no trade occurs in 1 second, all agents adjust their bids to improve the best existing bid. (2) The GD algorithm (Gjerstad and Dickhaut, 1998) submits orders to the market that maximize its expected surplus based on an updated belief distribution. The GD agent forms a belief about an offer or bid being accepted at price p based on the recent market history of accepted and unaccepted (including inframarginal and extramarginal) orders at that price. The authors find that different from past experimental studies on CDA markets with all-human design or all-algorithm design, the market price in their experiment shows slower convergence to the equilibrium price. Meanwhile, for both types of strategies, human traders underperform algorithms by about 20% in trading surplus.

rules. The following generations are offsprings of the young generations. Kirman (1993) suggests a related model of mutation of heterogeneous opinions, e.g., chartists and fundamentalists (see also Brock and Hommes (1997) and Lux and Marchesi (1999); Hommes(2006) surveys the literature on heterogeneous agents).

[4] Varying the share of Snipers and ZI agents, Brewer and Ratan (2019) find (in an all-algorithm setting) that market efficiency and Snipers' profits are strongly impacted when 20% or more of traders are Snipers.

6

Gjerstad (2007) studies how different paces of submitting bids and asks influence the trading performance of humans and the GD algorithm in a CDA market with induced values (Smith 1962). There are 6 buyers and 6 sellers in the experimental market. In the hybrid markets involving interaction between human and GD agents, 3 human buyers/sellers and 3 automated buyers/sellers are on each side of the market. Interested in reaction speed, the author differentiates between "patient" and "impatient" algorithmic traders regarding waiting time before submitting a new order. Patient traders submit bids and offer at a slower pace than impatient ones. The result of the paper shows that first, all markets achieve a very high level of efficiency (usually more than 99.5%).

Meanwhile, there seems to be a "curse of impatience" for algorithmic traders. If algorithmic buyers/sellers are too active in submitting new limit orders, they will push the price up/down and lower their profit. In general, the profit of patient algorithmic traders is highest, followed by the impatient ones, and human traders' profit is lowest. These results are obtained for the simplistic CDA market with induced values. [5]

In a more complex environment, in which earnings depend on the share price at period end, Feldman and Friedman (2010) study human-algorithm interaction in an experimental CDA market. Their experimental treatments vary the composition and the size of markets. Human traders interact with algorithmic traders in large markets (1 human and 29 robots or 5 human and 25 robots) and small markets (5 human traders and 5 robots). The key findings of their study include: (1) human traders' average trading gain is smaller than algorithmic traders, but they may outperform algorithmic traders in market crashes; (2) human traders tend to destabilize small markets and neither stabilize nor destabilize large markets; (3) human traders respond to the payoff gradient similarly as the algorithmic trader. In their study, it is interesting to note that human traders earned higher profits during crashes (i.e., lose less with extreme market volatility) and tend to sell faster after experiencing a loss, although generally exhibit the same trading behavior as the algorithms.

Tai et al. (2018) let one human subject interact in CDA markets populated with ZI traders or with adaptive algorithmic traders of SFDAT, including Kaplan's Sniper, GD, and ZIP. Surprisingly, subjects' earnings are higher in the treatment with adaptive algorithmic traders than with ZI traders. The authors conjecture that subjects' cognitive working memory capacity impacts their trading acuity and test this hypothesis in asymmetric and symmetric CDA markets of Smith (1962) type. The result confirms the hypothesis; subjects with high elicited working memory capacity earn higher profits than

[5] Algorithms seem to obtain better returns also in more complex environments (Sato et al. 2002), but the relative superiority of the algorithms can depend on the market conditions. That is a result of Sato et al. (2002) who report a hybrid human-algorithm market in which students interact with algorithmic traders programmed by teams of researchers at a conference.

those with low elicited working memory capacity; the difference is pronounced in the more complex environments, i.e., asymmetric markets and adaptive agents.

Akiyama et al. (2017) implement an algorithm that trades on fundamentals in a Smith et al. (1988) call-auction asset-market design involving belief elicitation on future prices. The authors propose two treatments to study the question of strategic uncertainty as a course for bubbles and crashes: treatment with 6 human subjects and treatment with 1 human trader and 5 algorithmic traders committing transactions at fundamental value. In the second scenario, strategic uncertainty is eliminated while participants have perfect information about the algorithm's presence and its performed strategy.

The results suggest that strategic uncertainty might partly explain observed mispricing in this market. Using the same experimental setting, Hanaki et al. (2018) show that traders' performance is negatively correlated with their confidence in their short-term price forecast. In a related study, Ahrens et al. (2019) also use this experimental design with the fundamentalist algorithm to investigate subjects' overconfidence in their price forecast to find that the level of overprecision (i.e., the narrowness of the predicted confidence interval) may be endogenously determined or influenced by the observed market price dynamics. It tends to go up (down) when the asset price goes up (down).[6]

2.3 Algorithm Speed

Faster than human response-time to profit from trading has been one of the main reasons for the adoption of algorithmic trading in asset markets, and therefore, it has been an innate research question how much the algorithmic trader profits from low latency i.e., the minimal response delay.

Das et al. (2001) vary the algorithm's response speed, introduced by a sleep-wake cycle, to examine the interaction between humans and algorithmic traders. The "fast" algorithm would be idle for 1 second and become active when a new quote or trade is made. The "slow" algorithm would be idle for 5 seconds and only become active when a trade is made. When active, the algorithm would update its orders by submitting a new order or updating the existing order. Das et al. (2001) find that both the "fast" and "slow" algorithms outperform their human counterparts and observe decelerating price trajectories in both set-ups.

[6] Besides the experimental studies studying the role of algorithm traders in financial markets, some studies employ algorithm traders and do not choose the impact of algorithm traders as the primary research question. For example, Cason and Friedman (1997) used algorithms trading at fundamentals to train subjects. In the learning to forecast experiment by Hommes et al. (2005), the authors also include a fundamental algorithmic trader in the market who constantly predicts and trades based on the asset's fundamental value. The purpose of including these robot traders is to mimic the mean-reverting forces in financial markets.

8

De Luca et al. (2011) adopt the "fast" and "slow" algorithm with slight adjustments. The "fast" algorithm can calculate and submits orders every 1 second, while the "slow" algorithm can only calculate every 2.5 seconds and submits orders every 10 seconds. It is interesting to note that the interactions between humans and algorithms are flipped in Das et al. (2001) and De Luca et al. (2011).[7] In Das et al. (2001), the majority of trades are observed in the "fast" algorithm setting, and in contrast, they are observed with the "slow" algorithm setting in De Luca et al. (2011). Algorithms tend to trade among themselves first before trading with human traders in the "slow" algorithm setting in Das et al. (2001) and in the "fast" algorithm setting in De Luca et al. (2011).

Cartlidge et al. (2012) conduct a series of laboratory experiments assessing the role of algorithms' speed in market efficiency and their performance in an environment where human traders and algorithms interact in the market. They demonstrate that the market inhabited with slower algorithms, whose trading speed resembles the speed of human traders, will be closer to a competitive equilibrium, and the market efficiency is enhanced. Also, in an environment where human traders interact with algorithms, Cartlidge and Cliff (2013, 2018) investigate the impact of the millisecond-by-millisecond speed of the stock price movement. They argue that there is a price movement speed threshold above which human traders can still engage in market transactions and trade with human traders and algorithms. Below the threshold, the speed is too fast for human traders to react, and they can no longer participate in the market. Essentially, it is a tipping point that creates a phase transition from a mixed human-robot phase to a robot-robot phase. Cartlidge and Cliff (2013, 2018) coined this phase transition as a robot-phase transition. They also show that faster algorithmic traders cause lower market efficiency and market disintegration so that algorithms interact with other algorithms instead of algorithmic traders interacting with human traders.

Peng et al. (2020) investigated the role of speed in different market structures and configurations. They employed hybrid continuous double auction markets where human traders exist side by side with algorithmic traders. They partially replicated the result of the study done by Das et al. (2001), which shows that algorithmic traders that employ simple and speedy adaptive trading rules could outperform human traders. In particular, they showed that the result of Das et al. (2001) only holds when these algorithmic traders act as buyers and the market is balanced in that the demand-supply schedules are symmetric and market structures are competitive.

2.4 Arbitrage Algorithms

In real-world exchanges, financial assets are traded in fragmented markets because the regulatory authorities seek to enforce competition among exchanges to avoid monopoly fees for transactions. Market fragmentation can lead to situations in which an identical asset is demanded or offered at different prices at different venues, thus creating an

[7] It is also important to note that Cartlidge and Cliff (2013) admitted a bug in the algorithm's code used in the experiments of Luca et al. (2011) and Cartlidge et al. (2012).

arbitrage opportunity (e.g., see Figure 1a). Algorithms can also provide arbitrage price discrepancies between an exchange-traded index fund and the assets that compose the index. Similar price discrepancies can arise with two or several different exchange-traded funds based on the same index or between a derivative financial contract and the underlying asset. Automation is usually much faster at exploiting arbitrage opportunities than manual transmissions and, therefore, arbitrage algorithms have been among the most frequently applied algorithmic traders in financial markets (Kirilenko and Lo 2013).

Harrison (1992) studies an 8-period lived asset with imperfect payoff information. Including two one-period-ahead futures markets, for period 4 and period 8, he implements an algorithm that arbitrages between spot and futures CDA market (in treatment 4). Harrison (1992) concludes that arbitragers could be crucial for ensuring the spot market's informational efficiency and help constrain the length of any mispricing in spot prices in the study.

Angerer et al. (2019) study algorithmic arbitrage in the setting of Charness and Neugebauer (2019), which allows for trading in twin markets of the Smith et al. (1988) type. The dividends in the two markets A and B are perfectly correlated modulo a shift, i.e., the B-share pays in each period the same dividend as the A-share plus a fixed payment of 24 cash units. The authors investigate two liquidity absorbing algorithms called FastBot (see Figure 1a) and SlowBot, the liquidity providing algorithm LiqBot (see Figure 1b), and the two control treatments NoBot (in which the potential participation of an algorithm is announced, but no algorithm participates) and Control (with no announcement and no algorithm). The FastBot arbitrager immediately exploits arbitrage opportunities in real-time when they arise, while the SlowBot arbitrager trades with a delay. The study suggests that algorithmic arbitrage improves market efficiency. The arbitrage algorithms help approximate the law of one price and marginally amend the discovery of the fundamental value. The market quality is generally enhanced. Volatility is lower, transaction volume higher, and, particularly in the LiqBot treatment, liquidity is enhanced relative to the NoBot treatment. The arbitragers reap some earnings from human subjects upon transaction by design.

Nonetheless, subjects' earnings are not significantly lower compared to the treatments without algorithms. Interestingly, the SlowBot algorithm amends market efficiency similar to the other two algorithms, although it earns only a fraction of what the other algorithms earn. Finally, the authors find no announcement effect (see the following subsection) comparing the treatments Control and NoBot.

10

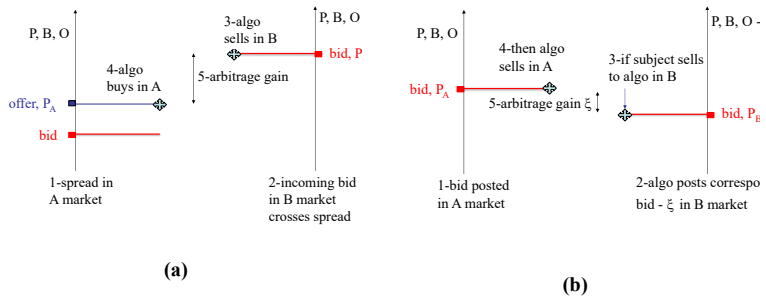


Figure 1. Liquidity (a) absorbing and (b) providing algorithmic arbitrageurs in Angerer et al. (2019)

P, B, O, F, ΔF, and ξ denote price, bid, offer fundamental value, the difference in fundamental value, and a random variable with support on the interval [0, F/2], respectively. The algorithmic trader exploits an arbitrage opportunity by selling high and buying low an identical claim of cash flows transacting at prices P_A and P_B. The sequence of events is numbered 1-4; 5 indicates the size of the arbitrage gain.

Neugebauer et al. (2020) test the Modigliani-Miller theorem of dividend policy irrelevance involving a FastBot algorithmic arbitrager (as in Angerer et al. 2019) and the trading of two 4- period lived assets in a complete asset market. Each asset pays a dividend at the end of the period, which is drawn without replacement from a set of four dividends. After the four regular dividends, shareholders receive a liquidating dividend which is high or low with equal probability. Owed to the fact that the remaining regular dividends are known, the difference in the fundamental value of the two assets is known in each period. Hence, if order in one market crosses the spread in the other market, an arbitrage opportunity arises (step 2 in Figure 1). In the treatment with the algorithmic arbitrager, such arbitrage opportunities are immediately exploited. The result of the study is that the law of one price (and thus dividend policy irrelevance) holds with and without arbitrager if dividend streams of both assets are identical. If dividends are not identical, the Modigliani-Miller theorem of dividend policy irrelevance can only be supported in the presence of (and must be rejected without) the algorithmic arbitrager. Hence, the result of the study adds further laboratory evidence that an algorithmic arbitrager may amend market efficiency.

Rietz (2005) studies index arbitrage in a 15-times repeated one-period CDA setting. At the beginning of each period, subjects are endowed with green and blue assets in a prediction market. One of the assets generates a dividend of $0.50 and the other a dividend of $0.00. The dividend- paying asset is determined by drawing from a bag with 14 green and 6 blue balls at the end of the trading period. Hence, the fundamental dividend value for the green asset is $0.35 and $0.15 for the blue asset, and predicted relative prices are $0.15/$0.35. During the period, subjects trade green and blue assets for cash, and subjects can buy a bundle containing one green and one blue asset from the experimenter or sell the bundle to the experimenter for the bundle's dividend value

of $0.50 in cash. Arbitrage opportunities arise whenever the sum of bids (offers) for the two assets totals more (less) than the bundle value. If such an opportunity arises, the arbitrager exchanges the bundle for the two assets. In the treatment with the arbitrager, subjects are informed about its functioning in the instructions. The results of the study are as follows: the arbitrager is involved in most of the transactions, and transaction volume and volatility increase significantly; prices drop relative to the treatment without arbitrager, where prices usually are above fundamentals; but relative prices are driven away from their predicted value, with prices of blue assets above and green assets below their fundamentals; and individuals holding less diversified portfolios. Hence, this evidence suggests that the arbitrager supports the law of one price but not always aids market quality and the price discovery of single asset fundamentals.

Grossklags and Schmidt (2006) also study arbitrage in a prediction CDA market. Differing from Rietz (2005), where the bundle involves two securities, Grossklags and Schmidt's bundle involves five securities and increased complexity. The algorithmic arbitrager is involved in about every fifth transaction. Surprisingly, price efficiency in terms of the law of one price does not increase with algorithmic arbitrage. Even more surprising, in one treatment, the algorithmic arbitrager's presence is not announced, and in that treatment, price efficiency is significantly worse than without the participation of the arbitrager.

Berger et al. (2020) study latency arbitrage in a repeated CDA market for a one-period lived asset with induced values, hence similar to Smith (1962) but with challenges to price discovery. In this setting, an algorithmic "HFT" trader, which is not announced, basically front-runs incoming orders to book an immediate gain. The first one, labeled directional trading algorithm, realizes an immediate gain via the submission of two orders within the queue when a subject submits a market order; for example, if two offers to sell are outstanding at 100 and 101 and a market buy order is submitted, the algorithm buys at the best offer of 100 and sells to the incoming bid at 100.9 just a point below the second-best offer price. The second one, labeled arbitrage algorithm, front-runs any incoming limit order that crosses the spread realizing an immediate gain (such as PB – PA in Figure 1a, but within one market) through two transactions. For example, if one offer to sell is outstanding at 100 and a bid at 101 is submitted to the market, the algorithm buys at the best offer of 100 and sells to the incoming bid at its limit of 101, thus realizing an immediate gain of 1 cash unit. Berger et al. (2020) report market quality enhancements, including an increase in transaction volume and bid-depth of the order book, in the human-algorithm environments relative to the baseline market with human subject only.

2.5 Manipulation

Market manipulations have always been a concern of market participants. Putniņš (2012) surveys manipulative practices in real-world exchanges, the theoretical and empirical literature. A great advantage of laboratory experiments on market manipulations compared to real-world discovery is that the experimenter can unequivocally identify manipulation in real-time and its effects of price distortion relative to fundamentals.

12

Leal and Hanaki (2018) address the HFT practice of market-making and the manipulative practice of spoofing in a CDA market with long-lived assets (Smith et al. 1988). " 'Spoofing' involves intentionally manipulating prices by placing an order to buy or sell a security and then canceling it shortly thereafter, at which point the spoofer consummates a trade in the opposite direction of the canceled order" (Kirilenko and Lo 2013, p. 66). Leal and Hanaki (2018) do not concentrate their analysis on the direct effects of spoofing and market-making but report the effects on subjects' beliefs of the potential presence of such an algorithm.[8] We report their experimental design and results in the following subsection.

Veiga and Vorsatz (2009, 2010) investigate the impact on price distortions from manipulation (similar to a "pump-and-dump" scheme, i.e., an attempt to boost the price of the stock to sell it high) performed through an algorithm in an experimental hybrid market. Veiga and Vorsatz (2009) set up an experimental CDA market for an asset that pays a high or low dividend with equal probabilities. The authors consider two treatments. In the control treatment and manipulation treatment, one-third of the market participants are informed with certainty about the dividend value. In the manipulation treatment, subjects know about the presence of the algorithm but not its strategy. The algorithm is programmed to buy 10 shares out of 24 when the market opens, thus pushing up the price, and then to sell them back to the market before the price returns to its normal level. The authors find that successful manipulation is possible when the asset's actual dividend value is low because there is a confusion between informed traders and manipulators. When the actual dividend value is high, the manipulator algorithm cannot distort prices because the competition between the informed traders ensures convergence to the dividend value.

In the follow-up study, Veiga and Vorsatz (2010) investigate manipulation in a CDA market with partially informed traders as in Plott and Sunder (1988). In this set-up, participants again trade one asset, taking three possible values with equal probabilities. In the first treatment, half of the participants are imperfectly informed about the asset's value (no aggregate uncertainty), while others are uninformed. In the second treatment, all subjects are partially informed, and in the third treatment, finally, only 1/3 of the participants are perfectly informed, with the others staying uninformed. The authors report that manipulation appears to be successful only with perfectly informed insiders when the asset's actual value is low and explains this result with the subjects' risk aversion. Veiga and Vorsatz's (2009, 2010) two laboratory experiments provide an argument in favor of the regulation obliging market insiders to disclose their transactions.

[8] The authors report data on the hybrid markets in the appendix of the paper. The transaction volume is increased relative to the markets without algorithms impacting mispricing and slowing convergence on fundamentals

Other experimental studies on manipulation do not use algorithmic traders but offer incentives to subjects to distort market prices (Hanson et al. (2006), Comerton-Forde and Putniņš (2011)).[9]

2.6 Announcement effect

Today, a person committing transactions in the financial market should reasonably expect an algorithmic trader as his or her counterparty. At the same time, the impact of an algorithm's presence or the possibility of its presence on humans' actions and expectations might be nontrivial. Thus, an important question regarding investor psychology is whether the possibility of interacting with an algorithm has a measurable influence on human behavior and the market. The evidence is mixed.

As pointed out above, Grossklags and Schmidt (2006) study a prediction market with an algorithmic arbitrager. The paper suggests that the announcement of the presence of algorithms increases market efficiency raising the rate of price convergence to the equilibrium relative to the setting where the algorithm is present, but this presence is not announced. Within the experiment, three treatments are investigated: no algorithm and no announcement (baseline); algorithm and no announcement; algorithm and announcement. Overall, announcement leads to the increased market efficiency, but at the same time, the algorithm's presence without announcement results in a decrease in the convergence rate in comparison to the baseline treatment. The authors explain that arbitrage algorithms tend to decrease the trading opportunities for humans, which results in a lower number of trades and distortion of the information aggregation process. However, when the presence of the algorithms is announced, subjects adapt their behavior by switching to more conservative trading strategies bidding closer to the fundamentals.

[9] Hanson et al. (2006) study price manipulation in a prediction market in which manipulator subjects receive a bonus payment based on price distortions. The authors find that the attempts to distort the price are short-lived due to successful counteractions of other market participants. However, it is important to emphasize that in this setting everyone was informed about the manipulators' presence, their objective function and the direction of the manipulation. Therefore, the authors call for further research with the relaxed assumptions before claiming that (prediction) markets cannot be manipulated. Comerton-Forde and Putniņš (2011) investigate the impact of closing price manipulation. One of the main findings is that market manipulations through aggressive buying or selling activities just before the market closing can effectively distort the price. They investigate the possibility of punishment of manipulators by the other market participants, but find that others are not always able to identify manipulation. In fact, despite closing price manipulation practices being illegal as they create an illusion of market interest and hinder the price discovery process, it seems complicated to actually proof and prosecute manipulations in financial markets.

Farjam and Kirchkamp (2018) also suggest a positive announcement effect. Their subjects seem to behave more rationally following the announcement, bringing transaction prices closer to the fundamental value than without the announcement. The experimental design involves a six- subjects CDA market with one multi-period lived asset (Smith et al. 1988). The study compares price deviations from the fundamental value across the two treatments: either subject is told that the algorithm may be present in their market or that the algorithm is not present. Meanwhile, no algorithm participates in the experiment. The authors align subjects' expectations by asking early participants to describe the algorithm and then sharing the prepared wordle with the other subjects claiming that the algorithm is programmed based on this description.[10]

Leal and Hanaki (2018) suggest no announcement effect on prices but find an effect in the elicited first-period beliefs. The experiment involves three treatments that differ in the instructions only. The treatment human-only (HO) makes no reference to algorithmic traders. In the instructions to the treatments spoofing (SP) and market-making (MM), subjects receive the information that they may interact with an algorithmic trader in the market, and the general strategy of the algorithms MM and SP are explained. SP is supposed to be taking advantage of human traders, while MM is supposed to provide more liquidity to the market. Surprisingly, the result of the experiment shows little difference between the two types of market. The results suggest that in MM and SP, relative to HO, initial average price forecasts are higher and more volatile. Initial orders are submitted later. Besides these effects, the market price in MM and SP deviates more from the fundamental value than in HO.

Finally, as pointed out above, Angerer et al. (2019) find no announcement effect and no pricing difference relative to fundamentals in the CDA market study with two perfectly correlated assets. The authors compare their control treatment without the announcement of potential algorithm participation with their NoBot treatment in which the potential participation is announced, but no algorithm participates. Different from the studies above, no information is disclosed on the strategy of the algorithm.

3 Algorithms in the hands of the subject

While many experiments sought to treat human traders independently from algorithmic traders, Aldrich and López Vargas (2020) and Asparouhova et al. (2020) allowed their subjects to choose to employ algorithmic strategies in-market experiments.

Aldrich and López Vargas (2020) asked subjects to choose a predefined market maker or sniper algorithm and decide on costly improvements in latency. In their experimental framework, a single asset is traded on a single exchange, and traders can submit limit orders to the exchange indicating the direction of trades, the limit quantity, the limit

[10] The authors also conducted a treatment in which an algorithm participates trading on fundamental value but report no data of that treatment.

price, and the duration with which the limit orders should remain active. Trades occur when a trader submits a market order that matches the highest (lowest) limit price listed in the order book. Aldrich and López Vargas (2020) consider two market environments: the CDA market and the first batch auction (FBA) format. In the experiment, trader-subjects can employ algorithms to conduct transactions on their behalf. The paper aims to compare the two market formats (CDA vs. FBA) in terms of a set of outcome variables, including market liquidity, traders' behaviors, the level of transaction costs, and informational efficiency. The paper shows that FBA is less prone to predatory trading behavior than the CDA.[11] In the CDA the algorithms produce permanent mispricing, and the authors report flash crashes in the first period.

Asparouhova et al. (2020) allow subjects to trade manually or deploy algorithms, and they are assumed to be aware of the potential presence of traders employing algorithms. The trading environment is a CDA market with the declining fundamental value of the underlying asset used in Smith et al. (1988). The algorithms either act as a market-maker or a reactionary bot. The market-maker bot provides liquidity by submitting a buy order (market-maker buyer) for one unit of an asset at 5 cents below the asset's fundamental value or sell order (market-maker seller) for one unit of an asset at 5 cents above the asset's fundamental value. The reactionary bot absorbs liquidity; it submits a buy order for one unit at fundamental value when a sell order arrives at 5 cents below the asset's fundamental value and submits a sell order at fundamental value when there is a buy order submission at 5 cents above the asset's fundamental value. Asparouhova et al. (2020) report that subjects utilize algorithms frequently, and roughly between 67%-80% of trades employed algorithms. They are interested in evaluating whether putting algorithms in the hands of subjects reduces the extent of asset mispricing but find no evidence to that effect. Price bubbles occur as frequently as without algorithms in the market. Further, they show that subjects who use algorithms do not earn higher earnings than manually trading subjects, and the use of algorithms causes a higher frequency of price surges in the first rounds of trading. For future research, it would be interesting to extend this framework to evaluate other types of algorithmic traders beyond market-making and reactionary algorithms.

[11] In a related study, Kahpko and Zoican (2020) investigate whether a speed-bump policy in a continuous auction environment could have a similar effect. The experimental treatments involve the submission of an order, the first arriving order wins or the winning order is chosen randomly if several orders arrive first. Orders generally arrive delayed. Subjects can make latency investments to decrease the arrival time. The authors find that subjects do invest in low-latency trading technology in the control treatment without speed bumps. In the experimental treatment, in which speed bumps artificially delay arrival times, investment in low-latency technology is not reduced if the speed bumps are identical to everyone. Only if speed bumps are heterogenous, investments in low latency technology drop by 20% relative to the control treatment. This result seems robust whether the time delay involved with the speed bumps is certain or uncertain.

16

4 Conclusion

The nascent experimental research on the interaction between human and algorithmic traders in experimental markets can be organized in studies in which algorithms are in the researcher's hands and those in which the researcher puts algorithms in the hands of the human subject.

In the first category, the algorithm in the researcher's hand, the reported studies have addressed research questions concerning the performance of algorithms and humans, the impact on market quality, and investor psychology. The answers to the questions are not unambiguous. The results suggest that algorithms (particularly the fast ones) frequently outperform humans in simple market settings. However, in more complex market situations, algorithms (particularly the fast ones) may do worse. Similarly, market quality would usually be enhanced in human-algorithm markets relative to all-human markets, particularly with passive algorithms like arbitragers, but may be worsened with manipulators. Investors' behavior and market prices may be attracted closer to fundamentals when the experimenter announces a possible interaction with an algorithmic trader, or no difference may be visible in the data. It seems to depend on the experimental design, and more data are needed to conclude.

In the second category, an algorithm in the hand of the subjects, real-world phenomena like flash crashes can be reproduced in the laboratory when strategies of inexperienced subjects align. According to the available studies, the efficiency of the CDA market may be unaffected if subjects take algorithms in their hand or if they trade by submitting orders. Again it would be good to have more data, possibly involving other algorithms than market makers and snipers.

To make a clear statement on the impacts of specific algorithms, we need replication studies. To better understand how hybrid markets work, all kinds of reasonable algorithmic trading systems should be studied in the laboratory, including those presented here, like SFDAT, genetic algorithms, etc., and algorithms not presented here, like neural networks. Besides, it might be interesting to conduct some studies with financial professionals as human subjects to understand better whether subjects with real market experience respond to the presence of algorithm traders or use the algorithm bots differently from "standard" participants in the laboratory experiments. As the one of Aldrich and Lopez Vega (2020), market design studies that experimentally evaluate trading institutions can be important to learn to avoid flash crashes.[12]

[12] An interesting paper in that respect is the simulation study of Brewer et al. (2013). The authors simulate limit orders at random arrival times in the CDA market to study market erosion amid a large order that introduces a flash crash. They find that the severity of the erosion depends on the market structure. Their study includes the following alternative structures; minimum resting times, trading halts, and switching to call auction mechanism amid a flash crash. Their results suggest that the temporal

To conclude, experimental hybrid market studies can be informative to researchers, traders, and regulators, whereas evidence from real-world observation is sometimes guesswork or sometimes impossible to obtain. We are at the beginning.

References

1. Ahrens, S., Bosch-Rosa, C., & Roulund, R. P. (2020). *Asset price dynamics and endogenous trader overconfidence.* Working Paper. Available at: https://www.macroeconomics.tu-berlin.de/fileadmin/fg124/Ahrens/Ahrens_BoschRosa_Roulund.pdf
2. Akiyama, E., Hanaki, N., & Ishikawa, R. (2017) It is not just confusion! Strategic uncertainty in an experimental asset market. *The Economic Journal, 127*, 563-580.
3. Aldrich, E. M., & Vargas, K. L. (2020). Experiments in high-frequency trading: comparing two market institutions. *Experimental Economics, 23(2),* 322-352.
4. Angerer, M., Neugebauer, T., & Shachat, J. (2019). *Arbitrage bots in experimental asset markets*. Working Paper.
5. Arifovic, J. (1996). The behavior of the exchange rate in the genetic algorithm and experimental economies. *Journal of Political economy*, *104*(3), 510-541.
6. Asparouhova, E. N., Bossaerts, P., Rotaru, K., Wang, T., Yadav, N., & Yang, W. (2020). *Humans in charge of trading robots: the first experiment*. Working Paper. Available at SSRN 3569435.
7. Beckhardt, B., Frankl, D. E., Lu, C., & Wang, M. I. (2016). *A survey of high-frequency trading strategies*. Working Paper. Available at: https://web.stanford.edu/class/msande448/2016/final/group5.pdf
8. Berger, N., DeSantis, M., & Porter, D. (2020). The impact of high-frequency trading in experimental markets. *The Journal of Investing*, *29*(4), 7-18.
9. Brewer, P. J. (2008). Zero-intelligence robots and the double auction market: a graphical tour. *Handbook of Experimental Economics Results*, *1*, 31-45.
10. Brewer, P., Cvitanic, J., & Plott, C. R. (2013). Market microstructure design and flash crashes: A simulation approach. *Journal of Applied Economics*, *16*(2), 223-250.
11. Brewer, P., & Ratan, A. (2019). Profitability, efficiency, and inequality in double auction markets with snipers. *Journal of Economic Behavior & Organization*, *164*, 486-499.
12. Brock, W. A., & Hommes, C. H. (1998). Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, 22(8-9), 1235-1274.
13. Cartlidge, J., & Cliff, D. (2013, February). Evidencing the "robot phase transition" in human-agent experimental financial markets. *Proceedings of 5th International Conference on Agents and Artificial Intelligence ICAART (1)*, 345-352.

switch to a call auction from the continuous double auction may be most effective in rebuilding the order book liquidity and recovery of the price level amid a flash crash. It would be interesting to confirm these effects in hybrid markets.

14. Cartlidge, J., & Cliff, D. (2018). Modelling complex financial markets using real-time human-agent trading experiments. In S-H. Chen, Y-F. Kao, R. Venkatachalam, & Y-R. Du (Eds.), *Complex Systems Modeling and Simulation in Economics and Finance* (pp. 35-69). (Springer Proceedings in Complexity). Springer International Publishing AG. https://doi.org/10.1007/978-3-319-99624-0_3
15. Cartlidge, J., Szostek, C., De Luca, M., & Cliff, D. (2012, February). Too fast too furious: faster financial-market trading agents can give less efficient markets. *Proceedings of 4th International Conference on Agents and Artificial Intelligence ICAART* (2), 126–135.
16. Cason, T., & Friedman, D., (1997). Price formation in single call markets. *Econometrica: Journal of the Econometric Society*, 65(2), 311-345.
17. Charness, G., & Neugebauer, T. (2019). A test of the Modigliani-Miller invariance theorem and arbitrage in experimental asset markets. *The Journal of Finance*, *74*(1), 493-529.
18. Cliff, D., & Bruten, J. (1997). Minimal-intelligence agents for bargaining behaviors in market-based environments. Technical report HP-97-91, Hewlett–Packard Research Labs, Bristol, England.
19. Comerton-Forde, C., & Putniņš, T. J. (2011). Pricing accuracy, liquidity and trader behavior with closing price manipulation. *Experimental economics*, 14(1), 110-131.
20. Das, R., Hanson, J. E., Kephart, J. O., & Tesauro, G. (2001, August). Agent-human interactions in the continuous double auction. *Proceedings of 17th International joint conference on artificial intelligence (IJCAI-01).* Lawrence Erlbaum Associates Ltd, 1, 1169-1178.
21. De Luca, M., Szostek, C. S., Cartlidge, J., & Cliff, D. (2011, September). *Studies of interaction between human traders and algorithmic trading systems.* Foresight Report - The Future of Computer Trading in Financial Markets, DR13.
22. Duffy, J. (2006). Agent-based models and human subject experiments. *Handbook of computational economics*, *2*, 949-1011.
23. Duffy, J., & Ünver, M. U. (2006). Asset price bubbles and crashes with near-zero-intelligence traders. *Economic theory*, *27*(3), 537-563.
24. Farjam, M., & Kirchkamp, O. (2018). Bubbles in hybrid markets: how expectations about algorithmic trading affect human trading. *Journal of Economic Behavior & Organization*, 146, 248-269.
25. Feldman, T., & Friedman, D. (2010). Human and artificial agents in a crash-prone financial market. *Computational Economics*, *36*(3), 201-229.
26. Gjerstad, S. (2007). The competitive market paradox. *Journal of Economic Dynamics and Control*, *31*(5), 1753-1780.
27. Gjerstad, S., & Dickhaut, J. (1998). Price formation in double auctions. *Games and economic behavior*, *22*(1), 1-29.
28. Gode, D. K., & Sunder, S. (1993). Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of political economy*, 101(1), 119-137.
29. Goldstein, M. A., Kumar, P., & Graves, F. C. (2014). Computerized and high-frequency trading. *Financial Review*, *49*(2), 177-202.

30. Grossklags, J., & Schmidt, C. (2006). Software agents and market (in) efficiency: a human trader experiment. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, *36*(1), 56-67.
31. Hanaki, N., Akiyama, E., & Ishikawa, R. (2018). Behavioral uncertainty and the dynamics of traders' confidence in their price forecasts. *Journal of Economic Dynamics and Control*, 88, 121 - 136.
32. Hanson, R., Oprea, R., & Porter, D. (2006). Information aggregation and manipulation in an experimental market. *Journal of Economic Behavior & Organization*, *60*(4), 449-459.
33. Harrison, G. W. (1992). Market dynamics, programmed traders and futures markets: beginning the laboratory search for a smoking gun. *Economic Record*, 68, 46-62.
34. Hommes, C. H. (2006). Heterogeneous agent models in economics and finance. *Handbook of computational economics*, *2*, 1109-1186.
35. Hommes, C., Sonnemans, J., Tuinstra, J., & Van de Velden, H. (2005). Coordination of expectations in asset pricing experiments. *The Review of Financial Studies*, 18(3), 955-980.
36. Khapko, M., & Zoican, M. (2020). Do speed bumps curb low-latency investment? Evidence from a laboratory market. *Journal of Financial Markets*, 55, 100601.
37. Kirilenko, A. A., & Lo, A. W. (2013). Moore's law versus murphy's law: algorithmic trading and its discontents. *Journal of Economic Perspectives*, 27(2), 51-72.
38. Kirilenko, A., Kyle, A. S., Samadi, M., & Tuzun, T. (2017). The flash crash: high-frequency trading in an electronic market. *The Journal of Finance*, 72(3), 967-998.
39. Kirman, A. (1993). Ants, rationality, and recruitment. *The Quarterly Journal of Economics*, 108(1), 137-156.
40. Leal, J. S., & Hanaki, N. (2018). *Algorithmic trading, what if it is just an illusion? Evidence from experimental financial markets*. Technical report, Groupe de REcherche en Droit, Economie, Gestion (GREDEG CNRS), Université Côte d'Azur, France.
41. Lux, T., & Marchesi, M. (1999). Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature*, 397(6719), 498-500.
42. March, C. (2019). *The behavioral economics of artificial intelligence: lessons from experiments with computer players*. BERG Working Paper No. 154.
43. Miller, R. S., & Shorter, G. (2016). *High frequency trading: overview of recent developments*. Congressional Research Service Report No. 4, Washington, DC.
44. Neugebauer, T., Shachat, J., & Szymczak, W. (2020). *A test of the Modigliani-Miller theorem, dividend policy and algorithmic arbitrage in experimental asset markets*. ESI Working Paper 20-14. Available at: https://digitalcommons.chapman.edu/cgi/viewcontent.cgi?article=1309&context=esi_working_papers
45. Peng, Y., Shachat, J., Wei, L., & Zhang, S. S. (2020). *Speed traps: algorithmic trader performance under alternative market structures*. ESI Working Paper 20-39. Available at: https://digitalcommons.chapman.edu/esi_working_papers/334/
46. Plott, C. R., & Sunder, S. (1988). Rational expectations and the aggregation of diverse information in laboratory security markets. *Econometrica: Journal of the Econometric Society*, 56(5), 1085-1118.

47. Putniņš, T. J. (2012). Market manipulation: a survey. *Journal of Economic Surveys*, *26*(5), 952-967.
48. Rietz, T. A. (2005). *Behavioral mis-pricing and arbitrage in experimental asset markets*. University of Iowa Working paper.
49. Rust, J., Miller, J. H., & Palmer, R. (1994). Characterizing effective trading strategies: insights from a computerized double auction tournament. *Journal of Economic Dynamics and Control*, 18(1), 61-96.
50. Sato, H., Matsui, H., Ono, I., Kita, H., Terano, T., Deguchi, H., & Shiozawa, Y. (2002). Case report on U-Mart experimental system: competition of software agent and gaming simulation with human agents. In Namatame A., Terano T., & Kurumatani K. (Eds), *Agent- Based Approaches in Economic and Social Complex Systems, Frontiers in Artificial Intelligence and Applications* (pp. 167-178). Tokyo, Japan: IOS Press & Ohmsha Ltd.
51. Smith, V. L. (1962). An experimental study of competitive market behavior. *Journal of political economy*, *70*(2), 111-137.
52. Smith, V. L., Suchanek, G. L., & Williams, A. W. (1988). Bubbles, crashes, and endogenous expectations in experimental spot asset markets, 56(5), *Econometrica: Journal of the Econometric Society*, 1119-1151.
53. Sunder, S. (2003). *Markets as artifacts: Aggregate efficiency from zero-intelligence traders*. Yale ICF Working Paper 02-16. Available at SSRN 309750.
54. Tai, C. C., Chen, S. H., & Yang, L. X. (2018). Cognitive ability and earnings performance: Evidence from double auction market experiments. *Journal of Economic Dynamics and Control*, 91, 409-440.
55. Treleaven, P., Galas, M., & Lalchand, V. (2013). Algorithmic trading review. *Communications of the ACM*, 56(11), 76-85.
56. Veiga, H., & Vorsatz, M. (2009). Price manipulation in an experimental asset market. *European Economic Review*, 53(3), 327-342.
57. Veiga, H., & Vorsatz, M. (2010). Information aggregation in experimental asset markets in the presence of a manipulator. *Experimental economics*, *13*(4), 379-398.

ProbLife: a Probabilistic Game of Life

Simon Vandevelde* and Joost Vennekens

KU Leuven, De Nayer Campus, Dept. of Computer Science
Leuven.AI - KU Leuven Institute for AI, B-3000 Leuven, Belgium
{s.vandevelde, joost.vennekens}@kuleuven.be

Abstract. This paper presents a probabilistic extension of the well-known cellular automaton, Game of Life. In Game of Life, cells are placed in a grid and then watched as they evolve throughout subsequent generations, as dictated by the rules of the game. In our extension, called ProbLife, these rules now have probabilities associated with them. Instead of cells being either dead or alive, they are denoted by their chance to live. After presenting the rules of ProbLife and its underlying characteristics, we show a concrete implementation in ProbLog, a probabilistic logic programming system. We use this to generate different images, as a form of rule-based generative art.

1 Introduction

Game of Life (or Life) [9] is a well-known cellular automaton invented by John Conway, which takes place in a rectangular grid consisting of cells that are either "dead" or "alive". The grid goes through multiple generations, simulating "evolution", in which cells can die, survive, or be born based on their number of living neighbours. It is often called a 0-player game: after selecting an initial state of cells, we sit back and watch the grid evolve through time.

In this paper we present ProbLife, which extends Game of Life with a probability element and continuous cell values in the range of [0..1]. While there already exist many extensions and variations of Life (and other cellular automata) that include probabilistic elements, each of these introduces the concept of "probability" in different ways. ProbLife distinguishes itself in two aspects: (a) rules in ProbLife have a probability associated to them and (b) instead of being limited to binary values, the cells can have any value in the continuous range of [0..1].

To play ProbLife, we create a practical implementation in the form of a probabilistic logic program in ProbLog. This approach allows us to elegantly represent the logic of ProbLife, in a flexible manner. As such, it can be used to quickly prototype different rulesets and experiment with the associated probabilities.

The act of generating grids from an initial state based on a predefined set of rules can be classified as a form of rule-based generative art [6]. While the grids generated by standard Game of Life can only be visualised dichromatically

* This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

(typically in black and white), the cells in ProbLife, due to their continuous nature, can be drawn using colour gradients.

In short, the contributions of this paper are:

- an overview of the Game of Life variants that include probabilities;
- the presentation of ProbLife, a probabilistic extension of Game of Life;
- a concrete implementation of ProbLife in ProbLog.

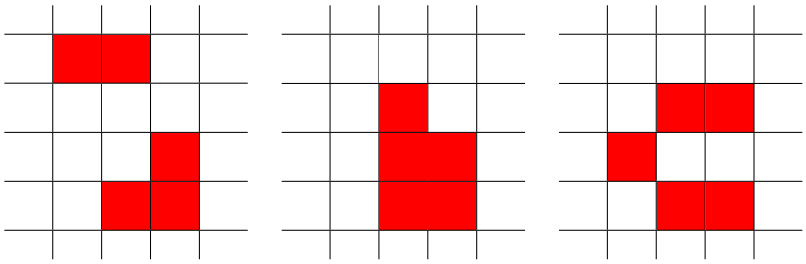
This paper is structured as follows. In Section 2, we elaborate on the specifics of Life and its variants, with a specific focus on those with probabilistic or continuous elements. Afterwards, we present ProbLife and its rules in Section 3, and show a concrete ProbLog implementation in Section 4. Finally, in Section 5 we discuss ProbLife in relation to the other probabilistic variants, present some interesting ProbLife instances we were able to find, and conclude.

2 Game of Life, extensions and variants

To play Conway's Game of Life [9], a player creates a state of living cells in a grid, after which they can observe the life inside evolve as defined by a set of rules. This set consists of two rules, which both depend on the exact number of living neighbours. The neighbourhood of a cell are those eight cells that directly surround it. The rules of Life are as follows:

1. A living cell survives to the next generation if it has exactly two or three living neighbours.
2. A cell is born if it was dead in the previous generation, and had exactly three living neighbours.

In this way, the first rule specifies the "survival" criterion for a living cell, and the second rule specifies the "birth" criterion for a dead cell to be born. An example of these rules in action is shown in Figure 1.



(a) Start state (b) State 2 (c) State 3

Fig. 1: Example of the Game of Life rules applied to a start state.

Many extensions and variants exist for Game of Life, which can be categorised based on how they differ from the original. The most straightforward variants

simply alter the ruleset, such as the cellular automatons "Flock", in which cells can only survive with 1 or 2 neighbours, and "Day and Night", which has four rules for survival, and four rules for being born. Other variants introduce more fundamental changes, such as converting Life into 3D [3], changing the size of the neighbourhood (e.g., Larger than Life [8]), changing the grid to be non-square [4], and replacing the rules of Life by a neural network that learned to "regrow" certain patterns [11].

For this paper however, we focus on those variants which add probability to the Game of Life – in any form whatsoever. For example, in [2] the authors describe their Life extension "Probabilistic Cellular Automata, Extension of the Game Of Life" (PCAEGOL), in which the value of a neighbouring cell can be "erroneous". Here, there is a certain probability of such an error occurring for each of the eight neighbours: for example, there could be a 20% chance to count the left neighbour as being dead, while it actually is alive. These errors are not consistent, in the sense that if a neighbour is considered erroneous for one cell, it might not be so for another cell. The added probability of errors leads to the game becoming nondeterministic, where the same initial state can lead to different outcomes.

In [10], the authors present Stochastic Game of Life (SGL). This variant adds probability in two aspects of the game: (1) the survival rules have a probability p_s associated to them, and (2) cells are always born if they have precisely three neighbours in the previous state, but also have a probability of p_b to be born if they have precisely two neighbours.

Some works, like that of [13], introduce a new stochastic component known as "temperature" T, which influences the probability of the rules in function of the *density* of the grid. Even further, regardless of the rules set by the player, T can influence the life or death of a cell, acting as a way of introducing chaos into the system.

In Life, and most other cellular automata, the value of all cells is updated *synchronously*, i.e., at the same time. [5] presents an asynchronous variant, where each cell is no longer guaranteed to be updated at each time step, but instead only has a chance to do so. Similar to the previous variants, this leads to a nondeterministic automaton.

Besides variants that introduce probability, there are also those variants that introduce continuous elements. For example, SmoothLife, as introduced in [12], transforms the rules of Life to work in a continuous grid, with a continuous function for the neighbourhood of a cell. In [1], the authors extend Life with continuous cell values, similar to this work. They model the game's rules using a continuous transition function, which also contains a temperature component T. As T rises, the transition function becomes less precise, which in turn causes the cell values to become increasingly fuzzy, representing "errors" in the system.

3 ProbLife

In ProbLife, the value of a cell is no longer restricted to 0 (dead) or 1 (alive). Instead, it can have any value of the continuous domain [0..1], where the value of a cell at time t represents the probability that the cell is alive at that time. For example, a cell value of 0.8 implies an 80% chance of living. Note that this preserves the meaning of 0 and 1 as guaranteed dead (0% chance to live) and guaranteed alive (100% chance to live). The value of a cell in ProbLife is defined by a set of rules that denote the probability of a cell surviving or being born, given its exact number of living neighbours. Such a rule, with a probability x and a number of living neighbours n is written as follows:

$$p_c(n) = x. \tag{1}$$

with n an integer between 0 and 8, c either "s" (survive) or "b" (birth), and x a real number between 0 and 1. For example, a rule stating that there is an 80% chance for survival with exactly 4 neighbours is denoted by the following rule.

$$p_s(4) = 0.8. \tag{2}$$

The value of a cell at column i, row j and time $t+1$ is then defined as:

$$C_{t+1}(i,j) = \sum_{n=0}^{8} N_t(i,j,n) \times \Big(p_s(n) \times C_t(i,j) + p_b(n) \times \big(1 - C_t(i,j)\big) \Big). \tag{3}$$

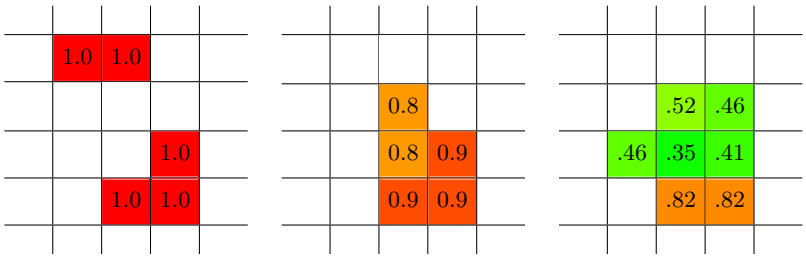
with $N_t(i,j,n)$ the probability that the cell at (i,j) had n neighbours alive at time t.

It is easy to see that ProbLife generalizes the original Game of Life, since we can recover the latter by the following ruleset:

$$\begin{aligned} &p_b(3) = 1. \\ &p_s(2) = 1. \\ &p_s(3) = 1. \\ &p_e(i) = 0, \text{ for all other } e \in \{s, c\} \text{ and } 0 \leq i \leq 8 \end{aligned} \tag{4}$$

Figure 2 shows an example of ProbLife in action. Here, the probability for a cell to live is shown in two ways: (1) the colour of the cell, where red represents a high probability, blue represents a low probability and green represents a probability in between, and (2) the number in the cell, which corresponds directly to its chance to live. The example uses a modified version of the standard Life ruleset, where the survival and birth probabilities have been set as respectively 90% and 80%: $p_s(2) = 0.9$, $p_s(3) = 0.9$, $p_b(3) = 0.8$. We have found this ruleset to give good results, and will continue to use it for all other examples in this work as well.

Due to the probabilistic nature of ProbLife, cell configurations often die out completely after a few generations. Indeed, on average the cell values will decrease with every generation, until the grid is empty. While there is no way to



(a) Start state (b) State 2 (c) State 3

Fig. 2: Example of ProbLife with $p_b(3) = 0.8$, $p_s(2) = 0.9$ and $p_s(3) = 0.9$.

reverse the decline, there are two main ways to get around this inevitable "extinction" by reaching stabilization. The straightforward solution is to add rules with a probability of 1 in such a way that the cells stabilize after a few generations. Alternatively, it is also possible to add a rule which causes dead cells with exactly 0 living neighbours to become alive (e.g. $p_b(0) = 0.8$), thereby turning ProbLife into a so-called "strobing rule[1]". In this latter case, the grid can never be empty for more than one generation, i.e., it is not possible that every cell has a zero probability of being alive for two consecutive states.

4 ProbLife in ProbLog

This section shows that ProbLife can be elegantly implemented in the ProbLog [7] system, a probabilistic extension of Prolog. This allows for quick experimentation with different rulesets as a way to easily create prototypes.

A ProbLog program consists of a set of probabilistic facts, and a set of rules. A probabilistic fact "$P_f :: f$" denotes a $P_f \in [0..1]$ probability for the atom f to be true. Rules in ProbLog are similar to those in Prolog, but with the addition of probabilities. Concretely, they are of the form

$$P_r :: h \ :- \ b_1, \ldots, b_n \tag{5}$$

where the *head* h evaluates as true with a probability of P_r if the *body* $b_1, \ldots, b_n$ evaluates as true. Here, the body consists of multiple *body atoms* b_i, which all need to be true for the body to be true. More information on the syntax and semantics of ProbLog can be found in [7]. We can now translate the rules of ProbLife to ProbLog as follows. A ProbLife rule $p_s(n) = z$ becomes:

$$z :: alive(X, Y, T) \ :- \ T > 0, T_p \ is \ T - 1, alive(X, Y, T_p), neigh(X, Y, T_p, N).$$

and a rule $p_b(n) = z$ is translated to:

$$z :: alive(X, Y, T) \ :- \ T > 0, T_p \ is \ T - 1, not(alive(X, Y, T_p)), neigh(X, Y, T_p, N).$$

[1] `https://conwaylife.com/wiki/Strobing_rule`

where $alive(X, Y, T)$ is a predicate to represent a cell being alive at position (X, Y) at time T, and $neigh(X, Y, T, N)$ a predicate to represent that the cell at (X, Y) has exactly N living neighbours at time T. To denote the set of initially alive cells, we also introduce a predicate $initAlive(X, Y)$, and add a rule that a cell is always alive at time 0 if it is part of the initial cells. For example, the ProbLife ruleset shown in (4) is represented in ProbLog as follows:

$$alive(X, Y, 0) \ :- \ initAlive(X, Y).$$
$$0.9 :: alive(X, Y, T) \ :- \ T > 0, T_p \ is \ T - 1, not(alive(X, Y, T_p)), neigh(X, Y, T_p, 3).$$
$$0.9 :: alive(X, Y, T) \ :- \ T > 0, T_p \ is \ T - 1, not(alive(X, Y, T_p)), neigh(X, Y, T_p, 2).$$
$$0.8 :: alive(X, Y, T) \ :- \ T > 0, T_p \ is \ T - 1, alive(X, Y, T_p), neigh(X, Y, T_p, 3).$$

Finally, the grid size and the initial state of a ProbLife instance are defined via ProbLog facts. E.g., the example shown in Fig. 2 contains the following facts:

$row(0).$	$row(1).$	$row(2).$	$row(3).$	$row(4).$
$col(0).$	$col(1).$	$col(2).$	$col(3).$	$col(4).$
$initAlive(1, 1).$	$initAlive(2, 1).$	$initAlive(3, 3).$	$initAlive(2, 4).$	$initAlive(3, 4)$

A full ProbLog implementation of ProbLife is available online[2], together with the code for a ProbLife editor.

5 Discussion & Conclusion

We will now briefly compare ProbLife to the other probabilistic variants described in Section 2. The main difference with the other automatons is that, while ProbLife introduces probability, the process of generating new states is actually deterministic. Indeed, instead of making a choice between setting a cell as alive or dead, we instead assign it the probability of being alive. When rerunning the same initial state, these living probabilities will always remain the same. Using ProbLog it is also possible to generate a sample run of a ProbLife system, i.e., one of the possible worlds. In this case the cells would be binary again, instead of continuous, and would represent one possible outcome. E.g., the Life example shown in Figure 1 is a sample of the ProbLife example shown in Figure 2. Moreover, it is possible to use ProbLife together with ProbLog's sampling feature to recreate the probabilistic automaton presented in [10].

A possible application of ProbLife is the modelling of uncertainty. E.g., consider a state in which there is a cell for which we are only 80% sure that it is alive. Using ProbLife, we could model the effect of this uncertainty on the rest of the population. Another application is the propagation of an infection on a population, using a modified version of ProbLife in which cells can be designated as sick, healthy and/or vaccinated. Using a specific ruleset (e.g., 80% chance of evolving in a sick cell if unvaccinated, but only 5% if vaccinated), we could model infections between generations.

[2] `https://gitlab.com/EAVISE/sva/ProbLife`

As mentioned earlier, generating images based on an initial state and a set of rules can be seen as a form of rule-based generative art. Using our ProbLog implementation, we experimented with many different initial states and rulesets in order to look for any interesting formations. The three most interesting ones of these are shown in Figures 3, 4 and 5.

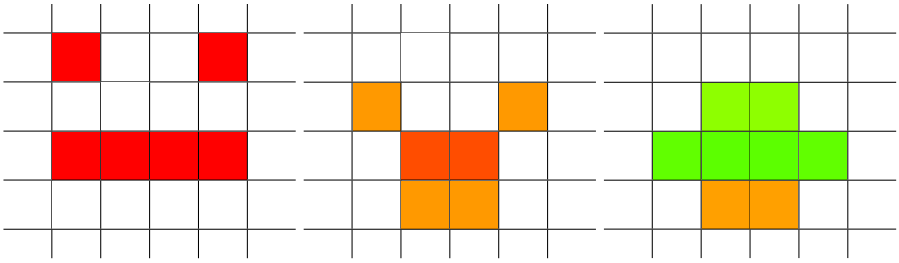


Fig. 3: "Unamused tree"

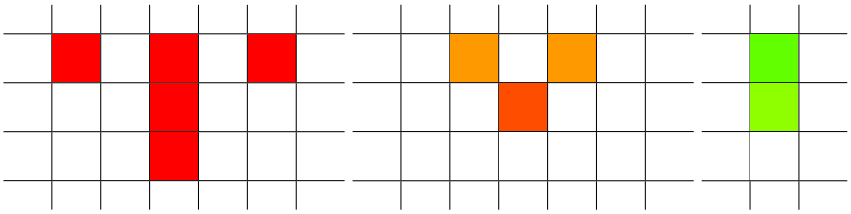


Fig. 4: "Reverse Butterfly", or, "Cold Water"

To conclude, this paper presents a probabilistic extension of Game of Life, called ProbLife. It distinguishes itself in the fact that its cells have continuous cell values in the range of [0..1], and that it remains deterministic. Each rule in ProbLife has a probability associated to it, meaning that it is possible for a rule not to be applied. Instead of the state of a cell being limited to either dead or alive, a cell in ProbLife is represented by its chance to live. We modelled a concrete implementation of ProbLife in ProbLog, as a way to straightforwardly experiment with different rulesets and initial states.

References

1. Adachi, S., Peper, F., Lee, J.: The Game of Life at finite temperature. Physica D: Nonlinear Phenomena **198**(3-4), 182–196 (2004), publisher: Elsevier
2. Aguilera-Venegas, G., Galán-García, J.L., Egea-Guerrero, R., Galan-García, M.A., Rodriguez-Cielos, P., Padilla-Domínguez, Y., Galán-Luque, M.: A probabilistic

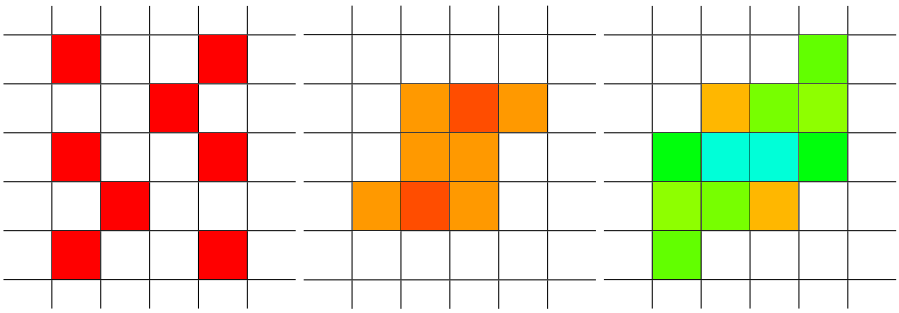


Fig. 5: “Fata Morgana” starts as heat lines and ends in a lush, green oasis

extension to Conway’s Game of Life. Advances in Computational Mathematics **45**(4), 2111–2121 (Aug 2019). https://doi.org/10.1007/s10444-019-09696-8, https://doi.org/10.1007/s10444-019-09696-8
3. Bays, C.: A new game of three-dimensional life. Complex Systems **5**(1), 15–18 (1991)
4. Bays, C.: The Game of Life in Non-square Environments, pp. 319–329. Springer London, London (2010), https://doi.org/10.1007/978-1-84996-217-9_17
5. Blok, H.J., Bergersen, B.: Synchronous versus asynchronous updating in the “game of life”. Physical Review E **59**(4), 3876 (1999), publisher: APS
6. Boden, M.A., Edmonds, E.A.: What is generative art? Digital Creativity **20**(1-2), 21–46 (2009)
7. De Raedt, L., Kimmig, A., Toivonen, H.: Problog: A probabilistic prolog and its application in link discovery. In: IJCAI. vol. 7, pp. 2462–2467. Hyderabad (2007)
8. Evans, K.M.: Larger than Life: It’s so nonlinear. The University of Wisconsin-Madison (1996)
9. Gardener, M.: Mathematical games: The fantastic combinations of john conway’s new solitaire game” life,”. Scientific American **223**(4), 120–123 (1970)
10. Monetti, R.A., Albano, E.V.: On the emergence of large-scale complex behavior in the dynamics of a society of living individuals: the stochastic game of life. Journal of theoretical biology **187**(2), 183–194 (1997), publisher: Elsevier
11. Mordvintsev, A., Randazzo, E., Niklasson, E., Levin, M.: Growing neural cellular automata. Distill (2020). https://doi.org/10.23915/distill.00023, https://distill.pub/2020/growing-ca
12. Rafler, S.: Generalization of conway’s ”Game of Life” to a continuous domain - SmoothLife (2011), arXiv: 1111.1567 [nlin.CG]
13. Schulman, L.S., Seiden, P.E.: Statistical mechanics of a dynamical system based on Conway’s game of Life. Journal of Statistical Physics **19**(3), 293–314 (Sep 1978). https://doi.org/10.1007/BF01011727, https://doi.org/10.1007/BF01011727

Trust Estimation in Forecasting-Based Knowledge Fusion*

Miroslav Kárný[1][0000−0002−7440−6041] and Daniel Karlík[1][0000−0001−8571−7534]

The Czech Academy of Sciences, Institute of Information Theory and Automation
18200 Prague 8, Czech Republic, school@utia.cas.cz, daniel@karlik.cz
https://www.utia.cas.cz/AS

Abstract. Inference and *d*ecision *m*aking (DM) are ultimate goals of the artificial-intelligence use. Complexity of DM tasks is the main barrier of their efficient solutions. Complex tasks are solved by dividing them among cooperating agents. This requires a knowledge fusion at a solution stage. It always has to cope with uncertainty. The used Bayesianism quantifies the uncertain knowledge by a *p*robability *d*ensity (pd) of modelled variables. The knowledge accumulation evolves the posterior pd of a parameter in the parametric model of observations. Bayes' rule updates the posterior pd. It provides a lossless compression of the knowledge in the observed data. An extended Bayes' rule enables the use of knowledge coded in a forecaster of the modelled observations supplied by an agent's neighbour. This rule exploits a weight expressing the trust into the forecaster. The paper offers yet-missing, algorithmic, data-based choice of this weight. It applies Bayesian estimation while assuming an invariant trust weight. Simulated examples illustrate behaviour of the resulting algorithm. They inspect its sensitivity to violation of the assumed credibility invariance. This prepares solutions coping with volatile knowledge sources.

Keywords: Trust · Knowledge sharing · Forecasting · Fusion · Decision making · Bayesianism.

1 INTRODUCTION

Complex decision-making (DM) tasks are solved by dividing them among cooperating agents[1], [7]. This requires a knowledge fusion at a solution stage, [33]. An agent locally models its environment. It selects its actions according to its local — in information space and time — aims. The efficiency of such an adaptive agent is enhanced (if not enabled at all) by sharing a knowledge with its neighbours in the information space. The neighbours are imperfect and may even act as adversaries. This makes the use of the shared knowledge strongly dependent on the *trust* assigned to neighbours. The trust quantification is actively studied in various contexts, [8,11,34], but it is far from being matured. The paper contributes to an improvement of this state. It deals with a specific, but well-applicable, knowledge-sharing scenario. The sharing supports an agent estimating a parametric model by using observations and Bayes' rule, [24]. Its

* Supported by MŠMT LTC18075 and EU-COST Action CA16228

[1] They are humans, technical tools and their mixed groups. The agent is referred by "it".

neighbour irregularly offers a forecaster of the same observation. It adds the number indicating how many data items the forecaster reflects. The agent processes them by the extended Bayes' rule. This rule has its origin in [15]. Its advanced, formally derived, versions are in [14,26]. They use a trust weight assigned to the neighbour.

The vital, but yet-unsolved, choice of the trust weight is addressed here.

Layout: Sec. 2 makes the paper self-reliant by recalling the used theory. Sec. 3 solves the addressed problem. Simulations in Sec. 4 illustrate the solution and inspect its sensitivity to the adopted invariance assumption. Sec. 5 touches the case of volatile credibility of the neighbour. Concluding remarks are in Sec. 6.

Notation: The text applies the next agreements:
$\{x\}$ is a set of x's, its nature is only revealed if need be; $|x|$ is cardinality of $\{x\}$;
$:=$ defines by assigning; $\propto$ is equality up to the normalisation; t marks discrete time;
✓ random variables, their values and realisations are formally undistinguished;
✓ models are *p*robability *d*ensities (pds[2]) marked by $\mathsf{sansmath}$ fonts as all mappings;
✓ functions with different arguments are different; the text prefers mnemonic labels;
$\mathsf{g}(x_t, y_{t-1}) := \mathsf{g}_t(x_t, y_{t-1})$: the time index of a function g drops if it is at its argument;
$\mathsf{p}_{t-1}(p)$ is the posterior pd of an unknown parameter $p \in \{p\}$, entering the parametric model; it is conditioned on the knowledge processed up to time $t-1$;
$\mathsf{p}(p|w, \mathsf{f}_t)$ enriches the condition of $\mathsf{p}_{t-1}(p)$ by the forecaster f_t with the trust weight w.

2 Preliminaries

An agent uses a parametric model $\mathsf{m}_t(o|r,p)$. This conditional pd relates the observation $o \in \{o\}$ to the regressors $r \in \{r\}$ and to an unknown parameter $p \in \{p\}$. The relation depends on time $t \in \{t\} := \{1, 2, \dots\}$. The posterior (conditional) pd $\mathsf{p}_{t-1}(p)$ quantifies the agent's knowledge about the unknown parameter $p \in \{p\}$ gained up to time $t-1$. Having data $d_t := (o_t, r_t)$, the pd $\mathsf{p}_{t-1}(p)$ updates by Bayes' rule, [24], to

$$\mathsf{p}_t(p) = \frac{\mathsf{m}(o_t|r_t,p)\mathsf{p}_{t-1}(p)}{\mathsf{m}(o_t|r_t,\mathsf{p}_{t-1})}, \quad \mathsf{m}_t(o|r,\mathsf{p}) := \int_{\{p\}} \mathsf{m}_t(o|r,p)\mathsf{p}(p)\,\mathrm{d}p, \quad t \in \{t\}. \tag{1}$$

The normalising pd $\mathsf{m}_t(o|r,\mathsf{p})$ models the observation o for the given regressors r and the knowledge about unknown parameter $p \in \{p\}$ stored in the pd $\mathsf{p}(p)$. It is agent's forecasting model. A subjective prior pd p_0, [29], starts the recursion (1).

In the inspected knowledge sharing, a neighbour provides to the agent its forecaster $\mathsf{f}_t(o)$ of the observations $o_t \in \{o\}$. This *non-normalised* pd should reflect the situation with the same regressors r_t as those used by the agent for forecasting of o_t. The number $\nu_t := \int_{\{o\}} \mathsf{f}_t(o)\,\mathrm{d}o \in (0, \infty)$ enhances the knowledge stored in the pd $\mathsf{f}_t(o)/\nu_t$. It declares the amount of data items used for creating the forecaster.

The neighbour forecasts using other knowledge resources than the agent. It means other models, theories, data sets, processing ways, expert's opinions, simulations, etc.

[2] Pd means Radon-Nikodým derivative, [28], i.e. both a probability density and mass function.

The theory we rely on, see Prop. 3 in [14], exploits the forecaster f_t by the extended Bayes' rule. It corrects the posterior pd $p_{t-1}(p)$ to the pd denoted $p(p|w_t, f_t)$

$$p(p|w_t, f_t) \propto p_{t-1}(p) \exp\left[w_t \int_{\{o\}} f_t(o) \ln[m(o|r_t, p)]\, \mathrm{d}o\right], \quad \text{where} \tag{2}$$

$w_t \in [0, 1]$ is the agent's trust weight assigned to the neighbour's forecaster f_t. The pd $p(p|w_t, f_t)$ is conditioned on the knowledge entering p_{t-1} enriched by the forecaster f_t weighted by w_t. The relation (2) indeed extends Bayes' rule as a fully trustable, $w_t = 1$, single, $\nu_t = 1$, crisp observation o_t is modelled by Kronecker's (Dirac's) pd

$$\delta(o, o_t) := \begin{cases} 1 & \text{if } o = o_t \\ 0 & \text{otherwise} \end{cases} \qquad \text{and reduces (2) to (1) as } p(p|w_t := 1, \delta_t) \overbrace{\propto}^{(2)}$$

$$p_{t-1}(p) \exp\left[1 \times \int_{\{o\}} \delta(o, o_t) \ln[m(o|r_t, p)]\, \mathrm{d}o\right] = p_{t-1}(p)[m(o_t|r_t, p)]^1 \overbrace{\propto}^{(1)} p_t(p).$$

It is practically important that for parametric models from *e*xponential *f*amily (EF, [4]), the functional rule (2) reduces to an algebraic updating of values of a sufficient statistic. EF consists of the parametric models of the form

$$m_t(o|r, p) := \exp\langle a_t(d), b(p)\rangle, \quad d := (o, r). \tag{3}$$

They are instantiated by multivariate functions a_t, b with their values entering the scalar product $\langle\cdot,\cdot\rangle$. In thought cases, the scalar product has the simple form

$$\langle a_t(d), b(p)\rangle := \sum_{i \in \{i\}} a_{ti}(d) b_i(p), \qquad |i| < \infty, \ \ t \in \{t\}, \tag{4}$$

where a_{ti}, b_i are known real-valued functions.

The used posterior pd p_t, conjugated to the model (3), [5], is given by the value of the $|i|$-dimensional statistic $\sigma_t = (\sigma_{ti})_{i \in \{i\}}$ with real-valued σ_{ti}. The pd reads

$$p_t(p) := c(p|\sigma_t) := \frac{\exp\langle \sigma_t, b(p)\rangle}{n(\sigma_t)}, \qquad n(\sigma) := \int_{\{p\}} \exp\langle \sigma, b(p)\rangle \, \mathrm{d}p < \infty. \tag{5}$$

Updating by the extended Bayes' rule (2) preserves the form (5). It holds

$$\begin{aligned} p_{t-1}(p) &= c(p|\sigma_{t-1}) \quad \overbrace{\Rightarrow}^{(2)} \quad p(p|w_t, f_t) = c(p|\sigma(w_t, f_t)) \\ \sigma_i(w_t, f_t) &= \sigma_{(t-1)i} + w_t a_i(f_t, r) \delta(r, r_t) \\ a_i(f_t, r) &:= \int_{\{o\}} f_t(o) a_{ti}(o, r)\, \mathrm{d}o, \ \ t \in \{t\}, \ \ i \in \{i\}, \ r \in \{r\}. \end{aligned} \tag{6}$$

This important case exemplifies the influence of the trust weight $w_t \in [0, 1]$.

Markov's chain as a member of EF: Markov's chain models the evolution of data with a finite number of possible values. Its parametrisation takes all transition probabilities as the unknown parameter. The next expression uses Kronecker's δ and $d = (o, r)$

$$p_{o_t|r_t} := \mathsf{m}(o_t|r_t, p) = \prod_{d\in\{d\}} p_{o|r}^{\delta(d,d_t)} = \exp\left[\overbrace{\sum_{d\in\{d\}} \underbrace{\delta(d, d_t)}_{\mathsf{a}_{o|r}(d_t)} \underbrace{\ln(p_{o|r})}_{\mathsf{b}_{o|r}(p)}}^{\langle \mathsf{a}(d_t), \mathsf{b}(p)\rangle}\right]. \tag{7}$$

This is an EF member (3), (4) with $i := o|r$. Its conjugated pd (5) is Dirichlet's pd $\mathsf{c}(p|\sigma) \propto \prod_{r\in\{r\}} \prod_{o\in\{o\}} p_{o|r}^{\sigma_{o|r}-1}$. The positive values of the statistic $\sigma := (\sigma_{o|r})_{o\in\{o\},r\in\{r\}}$ describe this pd. They enter the normalisation $\mathsf{n}(\sigma)$ (5), [13],

$$\mathsf{n}(\sigma) = \prod_{r\in\{r\}} \frac{\prod_{o\in\{o\}} \Gamma(\sigma_{o|r})}{\Gamma\left(\sum_{o\in\{o\}} \sigma_{o|r}\right)}, \quad \Gamma(v) := \int_0^\infty z^{v-1} \exp(-z)\, \mathrm{d}z, \quad v > 0. \tag{8}$$

The agent's forecasting model $\mathsf{m}_t(o|r, \mathsf{p})$ (1), found by (8) and $\Gamma(v+1) = v\Gamma(v)$, [1], is

$$\mathsf{m}(o|r, \mathsf{p}) = \mathsf{m}(o|r, \sigma) = \frac{\sigma_{o|r}}{\sum_{\tilde{o}\in\{o\}} \sigma_{\tilde{o}|r}}.$$

For $w_t \in [0, 1]$, $i = o|r$, $r, r_t \in \{r\}$, $o \in \{o\}$, the rule (6) gives the sufficient statistic

$$\sigma_{o|r}(w_t, \mathsf{f}_t) = \sigma_{(t-1)o|r} + w_t \mathsf{f}_t(o)\delta(r, r_t), \;\; o \in \{o\},\; r \in \{r\}.$$

3 Estimation of the Trust Weight

The unknown trust weight w_t in (2) is a hidden variable. Non-linear stochastic filtering, [10], estimates it optimally. It needs, however, the rarely-available time-evolution model and quite complex evaluations. This makes us to use local modelling, typical for adaptive systems. The inspected case of the *invariant trust*, $w = w_t, \forall t \in \{t\}$, prepares the general solution. Sec. 5 comments the volatile case.

The invariant w extends the parameter $p \in \{p\}$ to unknowns $(p, w) \in (\{p\}, [0, 1])$ entering the parametric model and the knowledge processing. A joint pd

$$\mathsf{p}_{t-1}(p, w) = \mathsf{p}_{t-1}(p|w)\beta_{t-1}(w) \tag{9}$$

describes the knowledge about (p, w) after time $t-1$ and before $t \in \{t\}$. The factorisation in (9) is the chain rule for pds, [24]. The conditional pd $\mathsf{p}_{t-1}(p|w)$ accumulates the knowledge about the unknown $p \in \{p\}$ when assigning the fixed trust weight w to the knowledge provided by the neighbour through forecasters offered before time t. The pd $\beta_{t-1}(w)$ expresses the agent's belief that w is the proper trust weight for the neighbour. The neighbour's forecaster $\mathsf{f}_t(o)$ enters the conditional version of (2)

$$\mathsf{p}(p|w, \mathsf{f}_t) \;\propto\; \mathsf{p}_{t-1}(p|w) \exp\left[w\zeta_t \int_{\{o\}} \mathsf{f}_t(o) \ln[\mathsf{m}(o|r_t, p)]\, \mathrm{d}o\right]$$

$$\zeta_t := \begin{cases} 1 & \text{if the forecaster } \mathsf{f}_t \text{ is available} \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

The introduced indicator ζ_t allows us to respect irregularity of processing of neighbour's forecasters without making the notation too complex. The agent's forecasting model, normalising (1) for the given $\mathsf{p}_{t-1}(p|w)$, is

$$\mathsf{m}(o|r, \mathsf{p}_{t-1}, w) := \int_{\{p\}} \mathsf{m}_t(o|r, p)\mathsf{p}_{t-1}(p|w)\,\mathrm{d}p. \tag{11}$$

In (10), (11), the weight w concerns the neighbour and thus it enters the posterior pds $\mathsf{p}_{t-1}(p|w)$ but not the agent's parametric model $\mathsf{m}(o|r, p)$.

The data-based updating of the belief $\beta_{t-1}(w)$ (9) into trust weights $w \in [0, 1]$ may realise after observing how much the neighbour's knowledge has contributed to the forecasting quality. The standard Bayes' rule gives, cf. (11),

$$\beta_t(w) \propto \mathsf{m}(o_t|r_t, \mathsf{p}_{t-1}, w)\beta_{t-1}(w). \tag{12}$$

The implementation of the recursion (10), (11), (12) is generally hard. It is simple for the discretised trust weight, [19]. The next proposition summarises such updating.

Proposition 1 (Parameter and Trust-Weight Estimation). *Let imminent trust weights be $w \in (w_k)_{k\in\{k\}}$, $\{k\} := \{1, \ldots, |k|\}$, $|k| < \infty$. They condition pds $\big(\mathsf{p}_{t-1}(p|w_k)\big)_{k\in\{k\}}$ quantifying the knowledge about the unknown parameter $p \in \{p\}$ of the pd $\mathsf{m}(o_t|r_t, p)$.*

The knowledge includes past data collected up to and including time $t-1$. It is enriched by irregularly available neighbour's forecasters with weights $(w_k)_{k\in\{k\}}$.

The values $(w_k)_{k\in\{k\}}$ express the neighbour's, supposedly invariant, credibility. They enter the updating of $\mathsf{p}_{t-1}(p|w_k)$ by the neighbour's forecaster f_t

$$\mathsf{p}(p|w_k, \mathsf{f}_t) \propto \mathsf{p}_{t-1}(p|w_k) \exp\left[w_k\zeta_t \int_{\{o\}} \mathsf{f}_t(o) \ln[\mathsf{m}(o|r_t, p)]\,\mathrm{d}o\right],\ p \in \{p\},\ k \in \{k\},$$

with $\zeta_t = 1$ if f_t is available and zero otherwise, cf. (10).

Let beliefs into trust weights w_k be $\beta_{t-1}(w_k)$, $k \in \{k\}$, see (9). Then, the updating of the pds β_{t-1}, p_{t-1} by data $d_t = (o_t, r_t)$ via the standard Bayes' rule reads, cf. (11),

$$\begin{aligned}
\beta_t(w_k) &= \frac{\mathsf{m}(o_t|r_t, \mathsf{p}_{t-1}, w_k)}{\mathsf{m}(o_t|r_t, \mathsf{p}_{t-1}, \beta_{t-1})}\beta_{t-1}(w_k)\\
\mathsf{m}(o|r, \mathsf{p}_{t-1}, w_k) &:= \int_{\{p\}} \mathsf{m}_t(o|r, p)\mathsf{p}_{t-1}(p|w_k)\,\mathrm{d}p\\
\mathsf{m}(o|r, \mathsf{p}_{t-1}, \beta_{t-1}) &:= \sum_{k\in\{k\}} \mathsf{m}(o|r, \mathsf{p}_{t-1}, w_k)\beta_{t-1}(w_k)\\
\mathsf{p}_t(p|w_k) &\propto \mathsf{m}(o_t|r_t, p)\mathsf{p}(p|w_k, \mathsf{f}_t).
\end{aligned} \tag{13}$$

Prop. 1 applied to EF (3) maps both Bayes' functional recursions to algebraic handling of the finite-dimensional statistic.

Proposition 2 (Estimation of Parameter and Trust Weight in Exponential Family). *Let trust weights $(w_k)_{k\in\{k\}}$ condition conjugated pds $\mathsf{p}_{t-1}(p|w_k) = \mathsf{c}(p|\sigma_{t-1}(w_k))$,*

(5). Let $(\beta_{t-1}(w_k))_{k\in\{k\}}$ be beliefs assigned to the trust weights. Their updating by the forecaster $f_t(o)$, preserves the conjugated form (5) and reads

$$\begin{aligned} p(p|w_k, f_t) &= c(p|\sigma(w_k, f_t)) = \frac{\exp\langle\sigma(w_k, f_t), \beta(p)\rangle}{n(\sigma(w_k, f_t))}, \quad n(\sigma) = \int_{\{p\}} \exp\langle\sigma, b(p)\rangle \, \mathrm{d}p \\ \sigma_i(w_k, f_t) &= \sigma_{(t-1)i}(w_k) + w_k \zeta_t a_i(f_t, r)\delta(r, r_t) \\ a_i(f_t, r) &:= \int_{\{o\}} f_t(o) a_i(o, r) do, \qquad r \in \{r\}, \ i \in \{i\}, \ k \in \{k\}, \end{aligned} \tag{14}$$

with ζ_t (10) respecting irregular availability of forecasters. The updating by the standard Bayes' rule, after having data $d_t = (o_t, r_t)$, see (6) and (14), reads

$$\sigma_{ti}(w_k) = \sigma_i(w_k, f_t) + a_i(d_t), \quad \beta_t(w_k) \propto \frac{n(\sigma(w_k, f_t))}{n(\sigma_{t-1}(w_k))}\beta_{t-1}(w_k), \ k \in \{k\}. \tag{15}$$

Thus, we have to store values of statistics $(\sigma(w_k), \beta(w_k))_{k\in\{k\}}$. The increments $a(f_t, r_t)$ (14) and $a(d_t) = a(\delta_t, r_t)$ (15) are evaluated once.

Trust estimation for Markov's chain: Specialisation of Prop. 2 and Sec. 2 imply that Dirichlet's pd is conjugated to the Markov's chain (7). Its degrees of freedom and beliefs into respective trust weights evolve, for $i = o|r$, as follows

$$\begin{aligned} \sigma_{o|r}(w_k, f_t) &= \sigma_{(t-1)o|r}(w_k) + w_k \zeta_t f_t(o)\delta(r, r_t) \\ \sigma_{(t)o|r}(w_k) &= \sigma_{o|r}(w_k, f_t) + \delta((o, r), (o_t, r_t)) \\ \beta_t(w_k) &\propto \frac{\sigma_{(t)o_t|r_t}(w_k)}{\sum_{o\in\{o\}} \sigma_{(t)o|r_t}(w_k)}\beta_{t-1}(w_k), \ \ (o, r) = d \in \{d\}, \ \ k \in \{k\}, \end{aligned} \tag{16}$$

where ζ_t (10) respects irregular offers of f_t.

Formulae (16) have strong intuitive appeal:

- the forecaster distributes its mass over possible observations $o \in \{o\}$ according to the probabilities $f_t(o)$ it assigns them, cf. quasi-Bayes techniques, [31];
- the agent attenuates f_t by the trust weight $w_k \in [0, 1]$ (discarding it for $w_k = 0$);
- the beliefs to weights reflect the neighbour's contribution to the forecasting quality.

The exploitation of the gained posterior pds depends on the DM task. For instance:

- a point estimate of the trust weight can be constructed, say, $\hat{w}_t := \sum_{k\in\{k\}} w_k \beta_t(w_k)$;
- Bayesian averaging may estimate parameter $p \in \{p\}$, say, via the marginal pd $p_t(p)$ $:= \sum_{k\in\{k\}} p_t(p|w_k)\beta_t(w_k)$ or similarly to forecast the observation $o_t \in \{o\}$ without specifying a point estimate of the weight;
- the trust estimate may serve to other, neighbour-related, inference or DM tasks.

4 Illustrative Experiments

Experiments illustrate the presented theory and show the sensitivity of the found estimator to the key assumption that the credibility of the neighbour's forecasters is invariant.

4.1 Simulation and Evaluation Conditions

The modelled environment was simulated by a discretised version of 2nd order auto-regressive-regressive Gaussian model

$$y_t = 1.9600y_{t-1} - 0.9604y_{t-2} + 0.0004a_t + 0.0004\varepsilon_t,$$

where ε_t was white zero-mean noise with unit variance; ε_t was independent of the past values $y_{\tau-1}, a_\tau$, $\tau \leq t$. The dynamics corresponds with the double real pole 0.98 and the unit static gains of actions and of the noise, [3]. Five-valued, integer, uniformly distributed, independent actions a_t were used, $|a| = 5$. A realisation of 10^5 samples, initiated by $y_0 = y_1 = 1$, was linearly mapped on positive values and discretised to ten-valued integer observations o_t, $|o| = 10$. The sequence $(o_t, a_t)_{t=2}^{10^5}$ was used for the choice of the *simulated* transition probability $p(o_t|o_{t-1}, o_{t-2}, a_t)$. The point estimate of this pd from the said realisation was used. Work [25] inspired this choice. The 2nd order Markov model was gained. The agent estimated 1st order model $p(o_t|o_{t-1}, a_t, p) = p_{o_t|o_{t-1},a_t}$, $r_t = (o_{t-1}, a_t)$, (7), i.e. the realistic mismodelling error was faced.

The neighbour's forecaster used the simulated transition probability with the inserted condition o_{t-1}, o_{t-2}, a_t. In the sensitivity-oriented experiments, this ideal forecaster was partially replaced by a randomly generated one, see below.

The trust-weight values $(w_k)_{k\in\{k\}} := \{0, 0.5, 1\}$, $|k| = 3$, Prop. 1, were inspected.

Prior statistics σ_0 (15) had randomly and independently assigned values 1 or 2.

Evaluations used 1000 *M*onte *C*arlo (MC) runs each lasting $|t| = 500$ steps, giving:

- Histograms of *beliefs* $\beta_t(w_k)$ (9) and of the *estimates*

$$\hat{w}_t := \sum_{k\in\{k\}} w_k \beta_t(w_k) \tag{17}$$

  of weights at the simulation end. Figures with time courses show their medians.
- Histograms of *forecast errors* per step compared to the best available forecast $\hat{o}_t^{\mathrm{i}}$ provided by the simulated transition probability

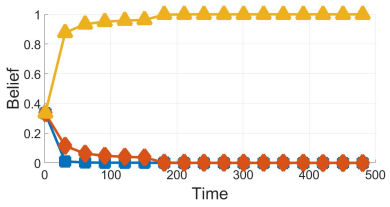
$$\Delta := \frac{1}{|t|} \sum_{t\in\{t\}} \Big| |o_t - \hat{o}_t| - |o_t - \hat{o}_t^{\mathrm{i}}| \Big|. \tag{18}$$

  There, o_t is the observation at the time t and judged $\hat{o}_t$ are the forecasts given by $m(o|r, p_{t-1}, w_k)$, $\forall\, k \in \{k\}$, (11) and by $m(o|r, p_{t-1}, \beta_{t-1})$ (13).
- Tables of basic *statistics* of the forecast errors (18) at the end of simulations. Their median, mean, standard deviation (STD) and root mean square error (RMS) are shown. RMS is taken as the primary indicator of quality when comparing the results.

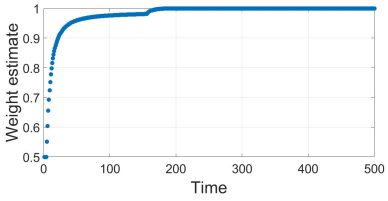
4.2 Invariant Ideal and Bad Neighbour's Forecasters

This part shows the behaviour of the proposed processing under met assumptions.

Ideal Neighbour's Forecaster: The neighbour's forecaster was the best possible one, i.e. the simulated $f_t(o) := p(o_t = o|o_{t-1}, o_{t-2}, a_t)$, $o \in \{o\}$, at realised o_{t-1}, o_{t-2}, a_t.



(a) Medians of beliefs in particular weights.

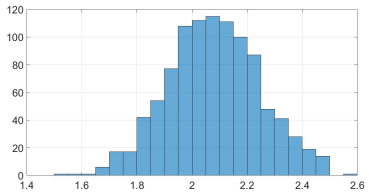


(b) Median of the weight estimates.

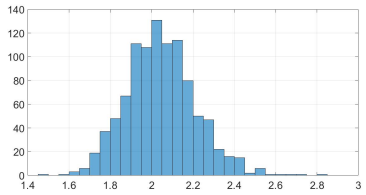
Fig. 1: shows medians of: (a) the beliefs $\beta_t(w_1)$ ■, $\beta_t(w_2)$ ♦, $\beta_t(w_3)$ ▲. (b) the weight estimate (17). It reflects 10^3 MC runs with the **ideal** neighbour's forecaster.

Results: Fig. 1 shows a fast convergence of the beliefs. The median of $\beta_t(w_3 = 1)$ raised rapidly to 1 and stayed there. Thus, the weight estimate (17) converged to 1, too.

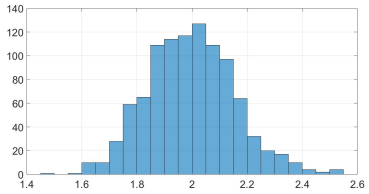
Fig. (2) shows histograms of forecast errors Δ (18). They are presented for completeness only. The differences are better seen on statistic values shown in Tab. 1.



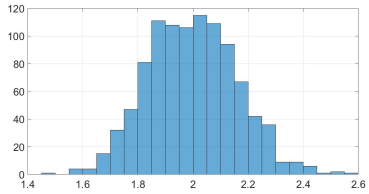
(a) Forecast errors of agent using $w_1 = 0.0$.



(b) Forecast errors of agent using $w_2 = 0.5$.



(c) Forecast errors of agent using $w_3 = 1.0$.



(d) Forecast errors of the proposed way.

Fig. 2: has counts of errors Δ (18) on the vertical axis and values of Δ on the horizontal axis. It reflects 10^3 MC runs with the **ideal** neighbour's forecaster.

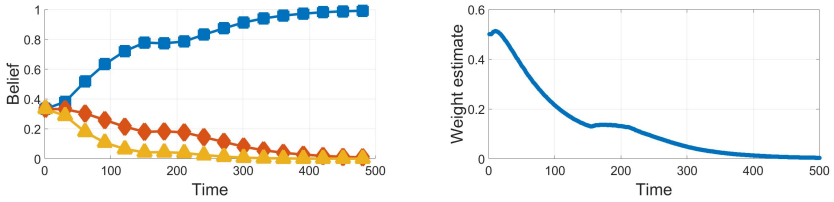
Discussion: The results confirm the desirable behaviour of the trust estimator. The high convergence rate is plausible. As predictable, the best quality is obtained for the fixed full weight assigned to the ideal forecaster. The proposed way is only slightly worse. The poorer performance is the cost for the lack of the knowledge of the proper weight.

Bad Neighbour's Forecaster: In this case, the neighbour's forecaster was chosen as useless as it was selected randomly without any relation to the simulated environment.

Table 1: Forecast errors Δ (18) with the **ideal** neighbour's forecaster.

Forecaster	Median	Mean	STD	RMS
Agent using $w_1 = 0.0$	2.076	2.079	0.167	2.086
Agent using $w_2 = 0.5$	2.038	2.035	0.165	2.042
Agent using $w_3 = 1.0$	1.992	1.996	0.159	2.002
Proposed way	1.997	1.998	0.161	2.005

Results: Fig. 3 shows that the proposed way behaves as desirable. The medians of beliefs into non-zero weights go quickly to 0. The point estimate $\hat{w}$ (17) goes also to 0.



(a) Medians of beliefs in particular weights. (b) Median of the weight estimates.

Fig. 3: shows medians of: (a) $\beta_t(w_1)$ ■, $\beta_t(w_2)$ ♦, $\beta_t(w_3)$ ▲. (b) the weight estimate (17). It reflects 10^3 MC runs with the **bad** neighbour's forecaster.

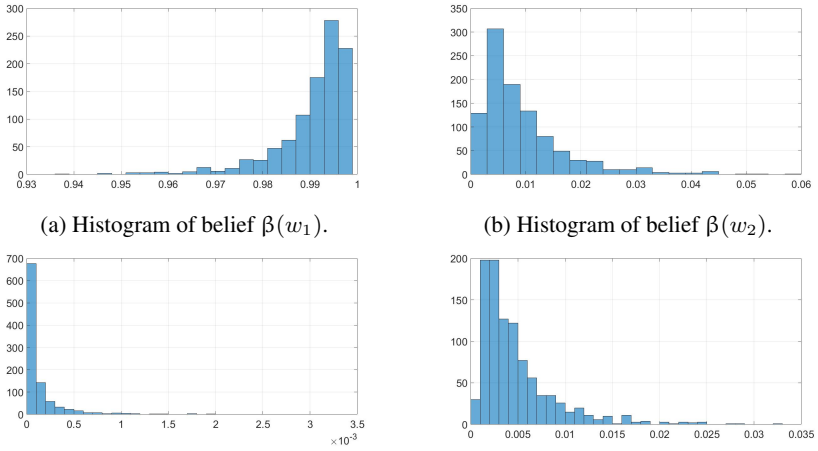
Histograms of forecast errors are poorly informative and they are left out. Their statistics are in Tab. 2. The best result is gained for the fixed zero weight ignoring the bad forecaster. The proposed way is close to it. It needed some data to recognise that the neighbour's forecaster is useless.

Table 2: Forecast errors Δ (18) with the **bad** neighbour's forecaster.

Forecaster	Median	Mean	STD	RMS
Agent using $w_1 = 0.0$	2.069	2.076	0.164	2.083
Agent using $w_2 = 0.5$	2.096	2.094	0.167	2.101
Agent using $w_3 = 1.0$	2.118	2.116	0.165	2.122
Proposed way	2.080	2.081	0.165	2.088

Fig. 4 complements the picture by presenting histograms of beliefs and the weight estimates (17) at the ends of simulation runs. They show quite small variations.

Discussion: The results confirm the expected desirable behaviour. Similarly as with the ideally forecasting neighbour, the poor forecasting was quickly recognised. As pre-



(a) Histogram of belief $\beta(w_1)$.

(b) Histogram of belief $\beta(w_2)$.

(c) Histogram of belief $\beta(w_3)$.

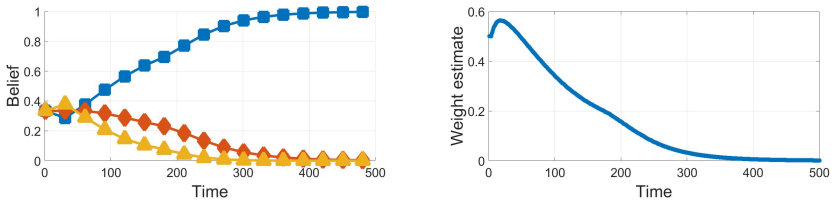
(d) Histogram of the weight estimate.

Fig. 4: shows counts of values on the vertical axis, the final values of $\beta_{|t|}$ and $\hat{w}_{|t|}$ on the horizontal axes. It reflects 10^3 MC runs with the **bad** neighbour's forecaster.

dictable, the best quality is obtained for the fixed zero weight assigned to the bad forecaster. The proposed way is only slightly worse. It pays for the lack of the knowledge.

4.3 Neighbour's Forecasters of Varying Reliability

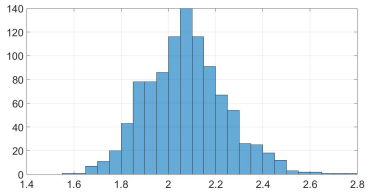
Randomly Failing Forecaster: In this experiment, the neighbour's forecaster consists of ideal forecasters in one half of randomly chosen time moments and of meaningless forecasters in the remaining half. The distribution of these choices were uniform. It is tempting to expect that the proper weight given to the forecaster will be around 0.5.
Results: Fig. 5 shows a small initial rise of the median of the belief $\beta_t(w_3)$. Since $t = 25$, it decreases to 0, which reached around $t = 400$. The median of the belief $\beta_t(w_2)$ behaves similarly. It leads to the weight estimates decreasing to 0.



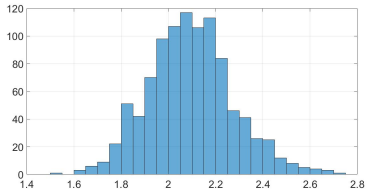
(a) Medians of beliefs in particular weights.

(b) Median of the weight estimates.

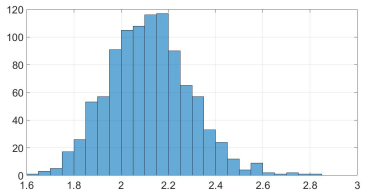
Fig. 5: shows medians of: (a) $\beta_t(w_1)$ ■, $\beta_t(w_2)$ ♦, $\beta_t(w_3)$ ▲. (b) the weight estimate (17). It reflects 10^3 MC runs with the **randomly failing** neighbour's forecaster.



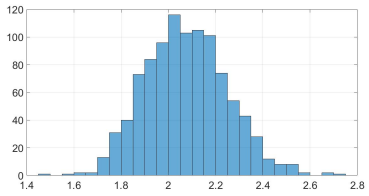
(a) Forecast errors of agent using $w_1 = 0.0$.



(b) Forecast errors of agent using $w_2 = 0.5$.



(c) Forecast errors of agent using $w_3 = 1.0$.



(d) Forecast errors of the proposed way.

Fig. 6: shows counts of errors Δ (18) on the vertical axis and the values of Δ on the horizontal axis. It reflects 10^3 MC runs with the **randomly failing** neighbour's forecaster.

Fig. 6 presents forecast errors. The only visible difference in Fig. 6 seems to be in Fig. 6d exhibiting a smaller amount of outliers. This might be a random effect so that statistics in Tab. 3 are more informative. Fig. 7 shows beliefs in the respective weights at the ends of simulation runs.

Table 3: Forecast errors Δ (18) with the **randomly failing** neighbour's forecaster.

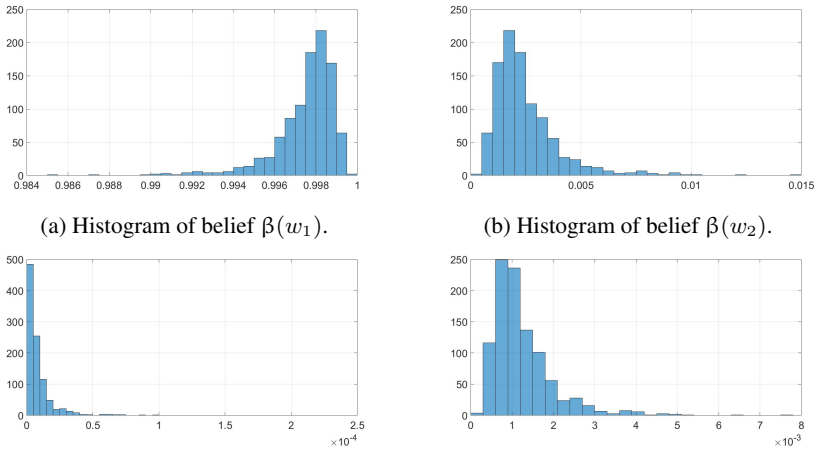
Forecaster	Median	Mean	STD	RMS
Agent using $w_1 = 0.0$	2.068	2.074	0.170	2.081
Agent using $w_2 = 0.5$	2.086	2.094	0.181	2.102
Agent using $w_3 = 1.0$	2.116	2.119	0.173	2.126
Proposed way	2.072	2.076	0.173	2.083

Discussion: Against the expectation, the ignoring of unreliable neighbour's forecaster is the optimal policy. The weight $w_1 = 0.0$ gives the best result. The proposed way converges to it giving the second best results.

Improving Forecaster: In this experiment, the forecaster begins with a bad quality and slowly throughout the simulation it is improving towards ideal reliability. Again, it is tempting to expect that the weight estimate $\hat{w}_t$ will converge to one.

Results: Fig. 8 shows a quite volatile evolution of beliefs. They oscillate before reaching (probably) stabilised values. The oscillations project into the weight estimate (17).

Tab. 4 summarises the forecast errors. It favourises to neglect the offered forecaster, $w_1 = 0.0$. The proposed way follows this and it is again the second best.



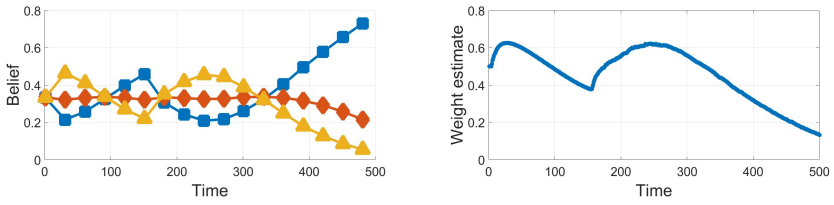
(a) Histogram of belief $\beta(w_1)$.

(b) Histogram of belief $\beta(w_2)$.

(c) Histogram of belief $\beta(w_3)$.

(d) Histogram of the weight estimate.

Fig. 7: shows counts of values on the vertical axis, the final values of $\beta_{|t|}$ and $\hat{w}_{|t|}$ on the horizontal axes. It reflects 10^3 MC runs with the **randomly failing** forecaster.



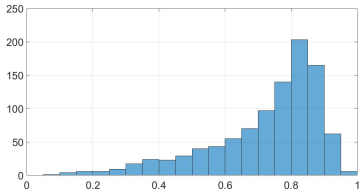
(a) Medians of beliefs in particular weights.

(b) Median of the weight estimates.

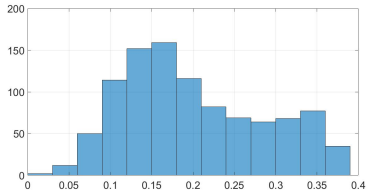
Fig. 8: shows medians of: (a) $\beta_t(w_1)$ ■, $\beta_t(w_2)$ ♦, $\beta_t(w_3)$ ▲. (b) the weight estimate (17). It reflects 10^3 MC runs with the **improving** neighbour's forecaster.

Table 4: Forecast errors Δ (18) with the **improving** forecaster.

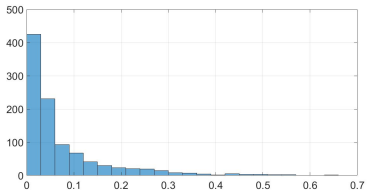
Forecaster	Median	Mean	STD	RMS
Agent using $w_1 = 0.0$	2.078	2.081	0.171	2.088
Agent using $w_2 = 0.5$	2.080	2.088	0.172	2.095
Agent using $w_3 = 1.0$	2.088	2.097	0.166	2.104
Proposed way	2.072	2.084	0.169	2.091



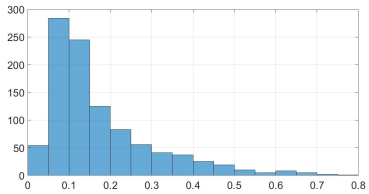
(a) Histogram of belief $\beta(w_1)$.



(b) Histogram of belief $\beta(w_2)$.



(c) Histogram of belief $\beta(w_3)$.



(d) Histogram of the weight estimate.

Fig. 9: shows counts of values on the vertical axis, the final values of $\beta_{|t|}$ and $\hat{w}_{|t|}$ on the horizontal axes. It reflects 10^3 MC runs with the **improving** neighbour's forecaster.

Fig. 9 confirms volatility of results in this scenario. It shows quite varying beliefs at the end of respective simulations.

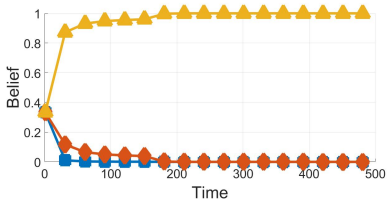
Discussion: The results discard the over-simplified expectation formulated above. The estimation dynamics and the forecaster-quality changes influence the results in a quite complex way. This confirms the need to relax the invariance assumption, see Sec. 5.

Deteriorating Forecaster: In this experiment, the neighbour's forecaster started as the ideal one and gradually deteriorated into the bad forecaster. The weight estimate (17) was expected to rise rapidly to 1 and then to decline to 0.

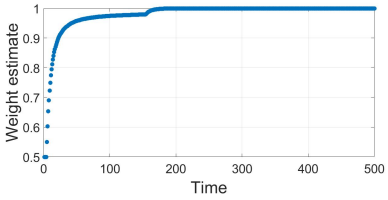
Results: Fig. 10 confirms the expectation for the initial phase but the weight estimate does not track the deterioration and stay close to 1 until the simulation end.

Tab. 5 evaluates the forecast errors and shows that the best results are gained when using fully the neighbour's forecaster all the time. The proposed way follows this pattern.

Discussion: The experiment confirmed that over-simplified expectations are violated when the estimation dynamics and the neighbour's forecaster with a varying reliability are combined. This makes the further progress outlined in Sec. 5 inevitable.



(a) Medians of beliefs in particular weights.



(b) Median of the weight estimates.

Fig. 10: shows medians of: (a) $\beta_t(w_1)$ ■, $\beta_t(w_2)$ ♦, $\beta_t(w_3)$ ▲. (b) the weight estimate (17). It reflects 10^3 MC runs with the **deteriorating** neighbour's forecaster.

Table 5: Forecast errors Δ (18) with the **deteriorating** forecaster.

Forecaster	Median	Mean	STD	RMS
Agent using $w_1 = 0.0$	2.076	2.081	0.173	2.088
Agent using $w_2 = 0.5$	2.043	2.046	0.168	2.053
Agent using $w_3 = 1.0$	1.999	2.006	0.163	2.013
Proposed way	2.002	2.009	0.161	2.015

5 Towards Handling Volatile Credibility

The assumed invariance of the estimated parameter fits to the assumed invariant trust weight. Adaptive systems [3] have a long tradition and experience how to cope with a slowly varying estimated parameter. Various types of forgetting (age-weighting) were proposed [22] and used even in connection with a trust handling [32].

The forgetting was recognised as a sort of flattening the evaluated posterior pds [16,17,18,23]. Thus, it can be directly applied both to p and w estimation, possibly using the idea of partial forgetting [6]. There are well-established rule of thumbs for the choice of forgetting factor. In critical cases, it may extend the estimated parameter, but it increases the computational complexity.

Possible abrupt changes of the estimated quantities were counteracted by adding a detector of such changes [9]. Recently, the problem was successfully and efficiently addressed by applying minimum *expected* relative principle [12]. Its tailoring to the discussed problem will be elaborated and published elsewhere.

6 Conclusions

Done: The paper contributes to a trustable knowledge sharing in a specific but widely applicable scenario. In it, a neighbour offers its forecaster of the observations handled by the supported agent. It complements the recent knowledge-sharing scenario [14] by the feasible estimation of the trust weight with which the neighbour's forecaster should be used. It primarily deals with the invariant weight quantifying the neighbour's credibility. The case fits the assumption that the parameter estimated by the agent is invariant. The performed, partially presented, experiments illustrate that the results are not

extremely sensitive to this assumption. The proposed solution, presented experiments and the discussion in Sec. 5 prepare general solutions for slowly as well as abruptly varying credibility of the neighbour's forecaster.

A comment on related works: The used knowledge-sharing way, complemented by the above trust learning, has unique inter-related features: ▶ it combines pds operating on *partially* overlapping domains, i.e. the agent and neighbour process the knowledge quite freely; ▶ the roles of the agent and the neighbour may swap, i.e. their mutual trust may even substantially differ. Such a support of agents cooperating without a mediator allows an unlimited scalability of the network interacting adaptive agents.

Future work: The need for the cooperation respecting credibility of the shared knowledge and the positive experience with the presented results make worthwhile to:

- ✓ perform extensive experiments delimiting the applicability range of the proposed technique, cf. no free lunch theorem, [35];
- ✓ apply the technique to important particular cases, say selected according to [11,34];
- ✓ elaborate the general solution to linear-in-unknown-parameter Gaussian model, [24], which is an important EF member suitable for modelling of dynamic environments with continuous-valued observations [27];
- ✓ extend the technique to other models like mixtures of EF members [20,21] requiring an approximate recursive estimation, [2];
- ✓ tailor the technique to other knowledge-sharing scenarios, up to an algorithm comparison [30], requiring an estimation of the trust weight [8];
- ✓ complete solutions coping with the volatile trustability.

You are invited to contribute to this important research. We are ready to cooperate and DK will share the relevant experimental software with you.

References

1. Abramowitz, M., Stegun, I.: Handbook of Mathematical Functions. Dover Publ., N.Y. (1972)
2. Andrýsek, J.: Estimation of Dynamic Probabilistic Mixtures. Ph.D. thesis, FNSPE CTU, 18208 Prague 8, Czech Republic (2005)
3. Åström, K., Wittenmark, B.: Adaptive Control. Addison-Wesley (1994)
4. Barndorff-Nielsen, O.: Information and Exponential Families in Statistical Theory. Wiley, N.Y. (1978)
5. Berger, J.: Statistical Decision Theory and Bayesian Analysis. Springer (1985)
6. Dedecius, K., Nagy, I., M.Kárný: Parameter tracking with partial forgetting method. Int. J. Adapt Control Signal Process. **26**(1), 1–12 (2012)
7. Dorri, A., Kanhere, S., Jurdak, R.: Multi-agent systems: A survey. IEEE Access **6**, 28573 – 28593 (2018)
8. Drawel, N.: Model Checking Trust-based Multi-Agent Systems. Ph.D. thesis, The Concordia Institute for Information Systems Engineering (2019)
9. Holst, J., Poulsen, N.: Self tuning control of plants with abrupt changes. IFAC Proc. Volumes **17**(2), 923 – 928 (1984), 9th IFAC World Congr.
10. Jazwinski, A.: Stochastic Processes and Filtering Theory. Ac. Press (1970)
11. Jiao, J., Zhou, F., Gebraeel, N., Duffy, V.: Towards augmenting cyber-physical-human collaborative cognition for human-automation interaction in complex manufacturing and operational environments. Int. J. of Production Research **58**(16), 5089–5111 (2020)

12. Kárný, M.: Minimum expected relative entropy principle. In: Proc. of the 18th ECC. pp. 35–40. Sankt Petersburg (2020)
13. Kárný, M., Böhm, J., Guy, T., Jirsa, L., Nagy, I., Nedoma, P., Tesař, L.: Optimized Bayesian Dynamic Advising: Theory and Algorithms. Springer, London, UK (2006)
14. Kárný, M., Hůla, F.: Fusion of probabilistic unreliable indirect information into estimation serving to decision making. Intern. Journal of Machine Learning and Cybernetics (2021). https://doi.org/10.1007/s13042-021-01359-9
15. Kracík, J., Kárný, M.: Merging of data knowledge in Bayesian estimation. In: Filipe, J., et al (eds.) Proc. of the 2nd Int. Conf. on Informatics in Control, Automation and Robotics. pp. 229–232. Barcelona (2005)
16. Kulhavý, R.: Restricted exponential forgetting in real-time identification. Automatica **23**(5), 589–600 (1987)
17. Kulhavý, R., Kraus, F.J.: On duality of regularized exponential and linear forgetting. Automatica **32**, 1403–1415 (1996)
18. Kulhavý, R., Zarrop, M.B.: On a general concept of forgetting. Int. J. of Control **58**(4), 905–924 (1993)
19. Lainiotis, D.: Partitioned estimation algorithms, I: Nonlinear estimation. Inf. Sci. **7**, 203–235 (1974)
20. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley Series in Probab. & Stat., Wiley, N.Y. (2000)
21. McNicholas, P.: Mixture model-based classification. CRC Press, Boca Raton, N.Y. (2017)
22. Milek, J., Kraus, F.: Time-varying stabilized forgetting for recursive least squares identification. In: Bányász, C. (ed.) IFAC Symp. ACASP'95, pp. 539–544. IFAC, Budapest (1995)
23. Peterka, V.: Subjective probability approach to real-time identification. In: The 4th IFAC Symp. on Identification and System Parameter Estimation, vol. 1, pp. 83–99. Tbilisi (1976)
24. Peterka, V.: Bayesian system identification. In: Eykhoff, P. (ed.) Trends & Progress in System Identification, pp. 239–304. Perg. Press (1981)
25. Podlesná, Y., Kárný, M.: Combination of forecasters in parameter estimation. Tech. Rep. 2385, ÚTIA AV ČR, Prague, Czech Republic (2020)
26. Quinn, A., Kárný, M., Guy, T.: Fully probabilistic design of hierarchical Bayesian models. Inf. Sci. **369**, 532–547 (2016)
27. Quinn, A., Kárný, M., Guy, T.: Optimal design of priors constrained by external predictors. Int. J. Approximate Reasoning **84**, 150–158 (2017)
28. Rao, M.: Measure Theory and Integration. J. Wiley (1987)
29. Savage, L.: Foundations of Statistics. Wiley (1954)
30. Schlender, T., Spanakis, G.: Thy algorithm shalt not bear false witness': An evaluation of multiclass debiasing methods on word embeddings. In: Baratchi, M., Cao, L., Kosters, W., Lijffijt, J., van Rijn, J., Takes, F. (eds.) BNAIC/Benelearn 2020. vol. 1398. Springer (2021)
31. Smith, A., Makov, U.: A quasi-Bayes sequential procedures for mixtures. J. of the Royal Statistical Society **40**(1), 106–112 (1978)
32. Vasilomanolakis, E., Habib, S., Milaszewicz, P., , Malik, R., Mühlhäuser, M.: Towards trust-aware collaborative intrusion detection: Challenges and solutions. In: Steghöfer, J., Esfandiari, B. (eds.) Trust Management XI. pp. 94–109. Springer Int. Publ. (2017)
33. Wang, P., Yang, L., Li, J., Chen, J., Hu, S.: Data fusion in cyber-physical-social systems: State-of-the-art and perspectives. Information Fusion **51**, 42 – 57 (2019)
34. Wang, Y.: Trust quantification for networked cyber-physical systems. IEEE Internet of Things Journal **5**(3), 2055–2070 (2018)
35. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Trans. on Evolutionary Computation **1**(1), 67–82 (1997)

Verbalizing but not just Verbatim Translations of Ontology Axioms

Vinu Ellampallil Venugopal[1] and P Sreenivasa Kumar[2]

[1] University of Luxembourg, Luxembourg, e-mail: vinu.venugopal@uni.lu
[2] Indian Institute of Technology, Madras, India, e-mail: psk@iitm.ac.in

Abstract. In this paper, we propose an inference-based technique to remove redundancy from natural language (NL) descriptions of Web Ontology Language (OWL) entities. The existing ontology verbalization approaches generate NL text segments that are closer to their counterpart statements in the ontology. Some of these approaches also perform grouping and aggregating of the text segments, aiming at a more fluent and comprehensive representation. However, we observed that the human-understandability of such descriptions is affected by the presence of repetitions and redundancies, and our studies show that such issues can be removed easily at the semantic level than at the NL level. We propose a novel technique called *semantic-level refinement* (or simply, *semantic-refinement*) for this purpose. Our approach aims at transforming the knowledge that is represented as a combination of less expressive (and not specific) logic-based expressions into the ones with high expressivity and specificity. This technique utilizes a predefined set of rules which are applied repeatedly on the restrictions associated with the individuals (and the concepts) to obtain a refined set of restrictions, guaranteed to be semantically equivalent to the original representation. Such refined sets of restrictions can then be verbalized to get concise descriptions of the ontology entities. Our experiments on ontologies from two different domains show that the proposed approach could significantly improve the readability of the NL texts when compared to the texts generated without a semantic-level refinement.

Keywords: Ontology Verbalization · Redundancy removal · Rule-based approach.

1 Introduction

Artificial Intelligence (AI) community widely uses *ontologies* for knowledge representation and reasoning. For example, the Gene Ontology[3] is now a very prominent resource in AI-powered Bioinformatics and Genomics. Another example is SNOMED CT[4], which is now fully formalized in OWL (Web Ontology Language) and widely used for electronic health records related applications. It is observed recently, that modeling knowledge in the form of ontologies helps to broaden the scope of cognitive AI and explainable AI (Peroni et al. (2008); Sarker et al. (2020)). However, the domain knowledge in the form of an ontology is inherently characterized by complex logical axioms, making the formalized knowledge not accessible to non-ontology communities (Dentler and Cornet (2015); E. Venugopal and Kumar (2020)). This had resulted in a large number of natural language (NL) verbalization tools for OWL ontologies

[3] http://geneontology.org/ [4] https://www.snomed.org/snomed-ct/

such as NaturalOWL (Androutsopoulos et al. (2014)) and SWAT Tools (Third et al. (2011)). However, the existing approaches in this direction mainly strive for one-to-one translation of logical constructs into the corresponding NL fragments. Such NL translations generally contain redundancies, as a domain concept could be expressed in several different ways in an ontology using the various constructs allowed in the ontology language—and, it is not guaranteed that one would always use the best combination to formalize the knowledge. In this paper, we explore a systematic approach that removes redundancies at the logic level—preserving semantic correctness—called *semantic-refinement*. And, it is found to be complementing the ontology verbalization application by generating concise NL sentences.

Motivating Example. Consider the following axioms from People & Pets ontology[5]:

(1) $\texttt{Cat\_Owner} \sqsubseteq \texttt{Person} \sqcap \texttt{Owner} \sqcap \exists \texttt{hasPet.Animal} \sqcap \exists \texttt{hasPet.Cat}$
(2) $\texttt{Cat\_Owner(sam)}$ (3) $\texttt{Cat} \sqsubseteq \texttt{Animal}$

The controlled natural language (CNL) descriptions for the individual `sam`, generated using standard OWL verbalizers, are as follows. From now on, we refer 'description' as the NL description of an entity (*individual* or *concept*) generated from the ontology.

- *A cat-owner is a person. A cat-owner is an owner. A cat-owner has as pet an animal. A cat-owner has a cat as pet. Sam is a cat-owner. All cats are animals.*
  or (with grouping and aggregation)
- *A cat-owner is a person and an owner . A cat-owner is all of the following: something that has pet an animal, and something that has a cat as pet; Example: sam. All cats are animals.*

As can be easily noted, these descriptions have redundant information, and attempting verbatim translation of each description logical (DL) construct has resulted in this situation. There are different types of redundancies one can observe here. The obvious type is the repetition of linguistically similar texts; e.g., "a cat-owner is an owner". The other type includes those generic restrictions which can be logically inferred from more specific restrictions; e.g., having said "A cat-owner has a cat as pet", it is not necessary to say "A cat-owner has as pet an animal." This paper deals with removing redundancies of the latter kind.

Contributions. In this paper, we propose a technique called *semantic-level refinement* (or simply *semantic-refinement*) that helps in removing the redundant (portion of the) restrictions and forms a more comprehensive description of an ontology entity. We particularly focus on generating descriptions from $\mathcal{SHIQ}$ DL ontologies. Our proposed approach generates descriptions of individuals and concepts by first representing the associated restrictions (knowledge) using a set of DL constructs that have high expressivity and high specificity than using a set that contains less expressive and generic expressions. If we revisit our previous example, we expect our approach to generate a text similar to "`sam` is an owner having at least one cat as pet"; such that the redundant portion of the text "has as pet an animal" is removed (since it clearly follows from

[5] http://www.cs.man.ac.uk/∼horrocks/ISWC2003/Tutorial/people+pets.owl.rdf

"having at least one cat as pet"). Due to page limitation, detailed proofs for the semantic correctness of the approach are made available in an extended version of the paper which we refer as *longer version*[6].

2 Related Work

Controlled Natural Languages. Over the last two decades, several CNLs such as Attempto Controlled English (ACE) by Kaljurand and Fuchs (2007), Rabbit by Hart et al. (2007), and Sydney OWL Syntax (SOS) by Cregan et al. (2007), have been specifically designed or have been adapted for ontology language OWL. All these languages are meant to make the interactions with formal ontological statements easier and faster for users who are unfamiliar with formal notations. Unlike the languages that were introduced to represent OWL in controlled English, proposed by Hewlett et al. (2005); Jarrar et al. (2006); Androutsopoulos et al. (2014), the aforementioned CNLs are designed to have formal language semantics and bidirectional mapping between NL fragments and OWL constructs. Even though these formal language semantics and bidirectional mapping enable a formal check to determine if the resulting NL expressions are unambiguous, they can result in generating a collection of unordered sentences that are difficult to comprehend. To use these CNLs as a means for ontology authoring and for knowledge validation purposes, the verbalized texts need to be properly organized. Stevens et al. (2011) have performed a detailed comparison of the systems that do such text organization. Among such systems, SWAT (Semantic Web Authoring) tools. are one of the recent and prominent tools which use standard techniques from computational linguistics to make the verbalized text more readable. They have tried to give better clarity to the generated text by grouping, aggregation, and elision. Third et al. (2011) have pointed out that the NL verbalization tools such as SWAT have given much importance to the linguistic fluency of the verbalized sentences than removing redundancies from their logical forms, and hence have deficiencies in interpreting the ontology contents.

Redundancy Removal. According to Alani et al. (2006), the works related to refining ontologies have focused only on ad-hoc application settings; not focusing primarily on preserving the semantics of the axioms. A notion for removing redundancies from ontologies without affecting the overall semantics, similar to what we propose in this paper, was proposed first by Grimm and Wissmann (2011). However, they have looked at redundancy in ontologies primarily from an ontology engineering and knowledge evolution point of view and were based on the notions introduced by Liberatore (2005) about redundant clauses in propositional logic formulas. Later, Third (2012) proposed a notion for removing redundancies in the context of ontology verbalization. In their work, the authors have established the fact that omitting "obvious axioms" while verbalization leads to a better reading experience for a human. By "obvious axioms" the author means those axioms whose semantics are in some sense obvious for an average human reader. For example, phrases such as "junior school" explicitly convey the meaning that a junior school is a school. In our work, we go further and establish that more inference-based redundancy removal could still be performed rather than just removing the morphological variants of the entity names, for greatly improving the quality and

[6] https://orbilu.uni.lu/retrieve/83875/90647/test.pdf

understandability of a verbalized text. Recently, Dentler and Cornet (2015) proposed four redundancy detection rules and the respective resolution methods, especially for SNOMED CT. However, there are no further efforts exist in generalizing such rules.

3 Preliminaries and Defintions

We assume that the readers are familiar with the semantics of $\mathcal{SHIQ}$ DL ontologies (Horrocks et al. (2000)). $\mathcal{SHIQ}$ DL is an extension of the well-known logic $\mathcal{ALC}$ (Schmidt-Schau and Smolka (1991)) with added support for role hierarchies, inverse roles, transitive roles, and qualifying number restrictions.

Running Example. In Fig. 1, we introduce a synthetic ontology called *academic (ACAD) ontology* which we follow throughout this paper as an example ontology.

TBox

IITStudent	≡	Student ⊓ ∀hasAdvisor.TeachingStaff ⊓ ∃hasAdvisor.Professor ⊓ ∃enrolledIn.IITProgramme
IIT_MS_Student	≡	IITStudent ⊓ ≤ 1 hasAdvisor.TeachingStaff
IITPhdStudent	≡	IITStudent ⊓ ≥ 2 hasAdvisor.TeachingStaff ⊓ ≤ 1 hasAdvisor.Professor
Professor	⊑	TeachingStaff
AssistantProf	⊑	TeachingStaff
⊥	≡	Professor ⊓ AssistantProf
⊥	≡	IIT_MS_Student ⊓ IITPhdStudent

ABox

```
IITStudent(tom)
IIT_MS_Student(tom)
hasAdvisor(tom, bob)
IITPhdStudent(sam)
hasAdvisor(sam, alice)
hasAdvisor(sam, roy)
AssistantProf(alice)
```

Fig. 1: TBox (Terminologies) and ABox (Assersions) of ACAD ontology

Label-set. The *label-set of an individual* is the set which contains *all* the class expressions and (existential, universal and cardinality) restrictions satisfied by that individual. A list of all label-sets from ACAD ontology is given in Table 1. The scope of the following formal definition of label-set is limited to $\mathcal{SHIQ}$ DL.

Definition 1. *Formally, the label-set of an individual x (represented as $\mathcal{L}_{\mathcal{O}}(x)$) is defined as: $\mathcal{L}_{\mathcal{O}}(x) = \{c_i \mid \mathcal{O} \models c_i(x)\}$ where c_i is of the following form: $c_i = A \mid \exists R.C \mid \forall R.C \mid \leq nR.C \mid \geq nR.C$. Here, A is an atomic concept, C is a class expression and R is a role name in ontology $\mathcal{O}$, and m and n are positive integers. C can be of the form: $C = A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists R.C_1 \mid \forall R.C_1 \mid \leq nR.C_1 \mid \geq nR.C_1$, where C_1 and C_2 are also class expressions.*

In the above definition, the c_is are free from disjunctions. If there exist a disjunctive clause satisfied by an individual, then the *satisfiablility* of each expression in that disjunctive clause should be checked and all such *satisfiable* expressions have to be included as conjuncts in the label-set. Clearly, then, the conjunction of all the elements in the label-set of an individual can be entailed by the ontology. That is, $\mathcal{O} \models \left(\sqcap_{i=1}^{n} c_i\right)(x)$. Here, the variable C will not be recursively expanded further to generate a large number of complex redundant expressions in the label-set. While this gives you a reasonable idea of how label-sets are generated, a more detailed account is presented in the longer version of the paper. Furthermore, the *label-set of a concept* can be defined as equivalent to the label-set of an individual that belongs to only that concept. Such a label-set

could be obtained easily by introducing a synthetic individual as the member of the concept and finding its label-set.

Table 1: Label-sets from ACAD ontology (intentionally omitted $\top$ class from the label-sets)

$\mathcal{L}_{\mathcal{O}}$(tom)	= { Student, IITStudent, IIT_MS_Student, ∃enrolledIn.IITProgramme, ≤1 hasAdvisor.TeachingStaff, ∀hasAdvisor.TeachingStaff, ∃hasAdvisor.Professor }
$\mathcal{L}_{\mathcal{O}}$(sam)	= { Student, IITStudent, IITPhdStudent, ∃isEnrolledIn.IITProgramme, ≥ 2 hasAdvisor.TeachingStaff, ≤ 1 hasAdvisor.Professor, ∀hasAdvisor.TeachingStaff, ∃hasAdvisor.Professor}
$\mathcal{L}_{\mathcal{O}}$(bob)	= { Professor, TeachingStaff }
$\mathcal{L}_{\mathcal{O}}$(alice)	= { AssistantProf, TeachingStaff }
$\mathcal{L}_{\mathcal{O}}$(roy)	= { Professor, TeachingStaff }

4 Proposed Verbalization Approach

Our verbalization process consists of three phases as shown in Fig. 2. The first phase takes an ontology as input and generates label-sets. In the second phase, we process these label-sets to a more refined form using our *semantic-refinement* technique—the main highlight of this paper. To understand the degree of reduction performed on a label-set, we assign a redundancy-score to the label-set while performing the reduction. Finally, we convert the restrictions in the refined label-sets into NL texts. In this section, we would first discuss the rationale for our refinement technique, and then we formally define the notion of semantic-refinement.

Fig. 2: Phases involved in the proposed verbalization method

Consider the label-sets of the individuals from ACAD ontology given in Table 1. A label-set would give us all the restrictions (logical expressions) that are satisfied by an individual. We can effectively verbalize all or part of these restrictions to frame a meaningful definition for that individual. For example, a well formed description for the instance tom that can be generated from its label-set is of the form: *"Tom is a student who is enrolled in an IIT Programme, has one professor as advisor, and all his advisors are teaching staffs."* Here, not all labels in the label-set were considered while generating the description. Some of the generic labels (mainly role restrictions) in the label-set if verbalized directly may generate confusing descriptions, and hence they should be reduced or combined with other restrictions (if possible) to get a more specific (so-called refined) restriction. For example, if left unrefined, the restrictions ∀hasAdvisor.TeachingStaff and ∀hasAdvisor.$\top$ may give rise to the description: *"all advisors are someone, and all advisors are teaching staffs"*, which may create ambiguity issues to a human reader. It is observed that to generate an unambiguous and a short description from a label-set, we have to identify redundant labels and see if they can be combined with the non-redundant labels to get a (highly expressive and more specific) refined form.

The naive method to perform the aforementioned tasks is by considering all combinations of labels and see if they can be reduced to a stricter form of logical expression. However, we could easily carryout this exhaustive process by considering labels of specific restriction types in a pre-defined order. For example, all the existential role restrictions could be considered prior to the universal role restrictions. Such a systematic process along with an ordered list of inference rules (called *refinement-rules*), that always generate stricter (more specific) forms of a given set of restriction, will ensure a fast refinement of the label-sets. Since we do this refinement of labels at the logical-level by considering their semantics, we call the refinement process as *semantic-refinement of label-sets*. The refined form of the label-set is called the *semantically-refined label-set*.

In addition to removing redundant labels in a label-set the semantic-refinement would also help in avoiding ambiguous verbalization of interim logical expressions. For example, the label: `∀hasAdvisor.Professor` can appear in the label-set of an individual of `IITStudent` due to the axiom: `IITStudent` $\sqsubseteq$ `∀hasAdvisor.Professor`. Linguistically, this label (along with the axiom) can be interpreted in two ways: either as *All advisors of IIT students are Professors* or, semantically, it can be interpreted as *Either all advisors of IIT students are Professors or* (vacuously-true case) *they do not have an advisor.* Clearly, considering the latter description, even though it is the semantically correct interpretation, may confuse a reader—especially the case when he could infer from other axioms that the vacuously-true case would not arise at all.

For identifying the cases where combinations of conditions involving qualifiers and/or number restrictions occur and to succinctly represent them, we introduce the following new constructors that have higher expressivity than the regular existential and universal restrictions.

- Non-vacuous role restriction: $\Im R.C$
  $\Im R.C^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}} \wedge \forall z.\langle x, z\rangle \in R^{\mathcal{I}} \implies z \in C^{\mathcal{I}}\}$
- Exactly-one role restriction: $\exists_{=1} R.C$
  $\exists_{=1} R.C^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | (\exists y_1.\langle x, y_1\rangle \in R^{\mathcal{I}} \wedge y_1 \in C^{\mathcal{I}} \wedge \exists y_2.\langle x, y_2\rangle \in R^{\mathcal{I}} \wedge y_2 \in C^{\mathcal{I}}) \implies y_1 = y_2\}$
- Exactly-n role restriction: $\exists_{=n} R.C$, general case of exactly-one role restriction.

In our semantic refinement process, like any rule-based approach, the order in which the inferencing rules are applied is also important as the applicability of one rule may depend on the other. We observed that there is a notion of *strictness* associated with role restrictions which can be effectively utilized for ordering the rules, such that the redundant label selection and the application of the rules can be done simultaneously. The notion of strictness can be looked at as: if a role restriction R_1 is implied by another role restriction R_2 (i.e., $R_2 \implies R_1$), then R_1 can be said to be a stricter version of R_2. For instance, $\Im R.U$ can be said as the stricter form of $\exists R.U$ and $\forall R.U$. Similarly, $\exists_{=n} R.U$ is a stricter form of $\leq nR.U$ and $\geq nR.U$. Since we intend to find stricter forms of role-restrictions, the obvious way is to apply rules corresponding to less strict restriction types prior to those of stricter restriction types. In general, the more strict-restrictions you have in the label-set more refined your label-set is. We can easily capture this notion by finding how often we apply the rules that do this refinement. To achieve this, we associate a pre-determined weight to each rule such that on applying a rule the overall *redundancy-score* of the label-set will reduce depending on

the weight of the rule. In other words, the objective of the semantic-refinement is to find the set which has the least redundancy score but yet guaranteeing the semantic-equivalence. The metric used for assigning the redundancy-score is detailed in the next section. The semantic-refinement of a label-set can be formally defined as:

Definition 2. *Given a label-set $\mathcal{L}_{\mathcal{O}}$ semantically-refined label-set can be defined as the set $\mathcal{L}_{\mathcal{O}}{}'$ such that $\forall\ x \in \mathcal{L}_{\mathcal{O}}$, $\exists\ y \in \mathcal{L}_{\mathcal{O}}{}' \mid y \models x$ (semantic equivalence) and in addition the set should have the least redundancy-score.*

Table 2: Details of rule sets 1-5.

Rule No.	*Restriction 1*	*Restriction 2*	*Condition*	*Refined form*
Concept Refinement rule				
1a	Concept names, whose (equality) definitions are already included in the label-set, can be removed.			
Superclass Refinement rule				
2a	U	V	$U \sqsubseteq V$	U
Existential Role Refinement rule				
3a	$\exists R.U$	$\exists S.V$	$U \sqsubseteq V\ \&\ R \sqsubseteq S$	$\exists R.U$
Universal Role Refinement rules				
4a	$\forall R.U$	$\forall S.V$	$U \sqsubseteq V\ \&\ S \sqsubseteq R$	$\forall R.U, \forall S.U$
4b	$\forall R.U$	$\forall R.V$	$V \sqsubseteq U$	$\forall R.V$
III & IV Combination rules				
5a	$\exists R.U$	$\forall R.U$		$\circlearrowleft R.U$
5b	$\forall R.U$	$\exists S.V$	$U \sqsubseteq V\ \&\ S \sqsubseteq R$	$\circlearrowleft R.U, \circlearrowleft S.U$
5c	$\forall R.U$	$\exists S.V$	$V \sqsubseteq U\ \&\ S \sqsubseteq R$	$\circlearrowleft R.U, \exists S.V$

5 Semantic-Refinement of Label-sets

We propose seven sets of rules for refining a label-set. Each of these rule sets contain carefully chosen rules which are repeatedly applied on the selected restrictions in the label-set until no more refinement is possible. More details of the algorithm follows.

Proposed Refinement Rules. The details of the first five sets of rules are given in Table 2. Each of the rule sets are named based on the type of restriction they handle. For example, the first rule set is called *Concept Refinement rule* since it refines the atomic concepts in the label-set.

- *Concept Refinement Rule (Rule 1a).* To apply this rule, we consider all the concept names that are present in the label-sets whose definitions (i.e., the set of restrictions which defines the concept) already included in the label-set. If the set of restrictions defining a concept completely exists in the label-set, then that concept name could be treated as a redundant information and shall be removed.
- *Superclass Refinement Rule (Rule 2a).* The label-set of an individual contains all the concept names which it belongs to. Some of the concepts in these label-sets are hierarchically related (in class - super-class relationship) in the ontology, resulting in redundant labels. For example, consider the label-set $\mathcal{L}_{\mathcal{O}}(\texttt{tom})$ in Table 1,

it contains the concepts `IIT_MS_Student` and `IITStudent`. Since it can be inferred from the concept `IIT_MS_Student` that `tom` is also a `IITStudent`, we can say that `IITStudent` is a redundant information (label) in the label-set. We remove such redundant labels by preserving only the most-specific concept. If the most specific concept had been already removed by Rule 1a, the next most specific concept name would be preserved in the label-set using this rule.

- *Existential Role Refinement rule (Rule 3a).* We can select two labels of the form: $\exists R.U$ and $\exists S.V$, from the label-set, as candidates for applying this rule, if $U \sqsubseteq V$ and $R \sqsubseteq S$, in the ontology. According to the existential role refinement rule, candidate labels are semantically equivalent to stating only a single restriction of the form $\exists R.U$ (which we call as the *refined form* of the labels). In general, all the rules that we cover in this paper are defined such that given a refined form and the condition which have been used for refinement, the non-refined forms of the restriction(s) could be traced back. That means, the refinement is done without affecting the semantics/meaning of the restrictions. The formal proofs of all the rules could be found at the longer version of the paper.
- *Universal Role Refinement rules (Rules 4a & 4b).* This rule set contains two rules that refine universal role restrictions.
- *III & IV Combination rules (Rules 5a, 5b & 5c).* For applying the rules in this rule set, we select combinations of existential and universal role restrictions from the label-set. The rules help in refining such combinations to a reduced form.

Table 3: Details of rule sets 6 and 7.

Rule No.	*Restriction 1*	*Restriction 2*	*Condition*	*Refined form*
Qualified Number Restriction Refinement rules				
6a	$\geq nR.U$	$\geq mS.V$	$U \sqsubseteq V$ & $R \sqsubseteq S$ & $n \geq m$	$\geq nR.U$
6b	$\exists R.U$	$\geq nS.V$	$V \sqsubseteq U$ & $S \sqsubseteq R$ & $n \geq 1$	$\geq nS.V$
6c	$\exists R.U$	$\leq nR.V$	$U \sqsubseteq V$ & $n = 1$	$\exists_{=1} R.U, \exists_{=1} R.V$
6d	$\geq nR.U$	$\leq nS.V$	$R \sqsubseteq S$ & $U \sqsubseteq V$	$\exists_{=n} R.U, \exists_{=n} S.V$
Exactly-n Role Refinement rules				
7a	$\exists R.U$	$\exists_{=1} S.V$	$U \sqsubseteq V$ & $R \sqsubseteq S$	$\exists_{=1} R.U, \exists_{=1} S.V$
7b	$\Im R.U$	$\exists_{=1} S.V$	$U \sqsubseteq V$ & $R \sqsubseteq S$	$\exists_{=1} R.U, \exists_{=1} S.V, \Im R.U$
7c	$\geq mR.V$	$\exists_{=n} R.U$	$U \sqsubseteq V$ & $m \geq n$	$\exists_{=n} R.U, \geq (m-n) R.(V \sqcap \neg U)$

The details of the next set of rule sets are given in Table 3.

- *Qualified Number Restriction Refinement rules.* In this set there are four rules. Here we mainly try to refine qualified number restriction restrictions (of the form $\leq nR.U$ or $\geq mS.V$) to stricter version of the same form or to a exactly-n restrictions.
- *Exactly-n Role Restriction rules.* In this rule set, we reduce the exactly-n role restrictions which are generated using the preceding rule-sets.

Algorithm for Semantic-Refinement. As we mentioned before, semantic-refinement helps in refining restrictions in a label-set to their stricter forms by combining them

Algorithm 1 Semantic-refinement of label-sets

procedure SEMANTIC_REFINEMENT($\mathcal{L}_{\mathcal{O}}$(x))
  Mark all $u \in \mathcal{L}_{\mathcal{O}}(x)$ as not PRs
  Apply *Concept Refinement rule* and remove appropriate concept names from $\mathcal{L}_{\mathcal{O}}$(x)
  $R \leftarrow$ Rule-sets 2-7 ▷ list of pre-defined rules
  for each rule-set $rs \in R$ **do**
    Let $M, REF \leftarrow \phi$
    for each $(u, v) \in \mathcal{L}_{\mathcal{O}}(x) \times \mathcal{L}_{\mathcal{O}}(x)$ AND $u \neq v$ **do**
      If !MARKED_AS_PR(u) AND !MARKED_AS_PR(v) **then**
      for each $(r \in rs)$ **do**
        if r is applicable on (u, v) **then**
          $M \leftarrow$ APPLY_RULE(r, u, v)
          $\mathcal{L}_{\mathcal{O}}(x) \leftarrow \mathcal{L}_{\mathcal{O}}(x) \cup M$
          $REF \leftarrow REF \cup$ {u,v}
          if $u \in M$ **then**
            $REF \leftarrow REF \backslash \{u\}$
          end if
          if $v \in M$ **then**
            $REF \leftarrow REF \backslash \{v\}$
          end if
        end if
      end for
      end if
    end for
    MARK_AS_PR(REF)
    $\mathcal{L}_{\mathcal{O}}(x) \leftarrow \mathcal{L}_{\mathcal{O}}(x) \cup REF$
    for each $u \in \mathcal{L}_{\mathcal{O}}(x)$ **do**
      if the restn. type of u is not used in the successive rule-sets AND MARKED_AS_PR(u) **then**
        $\mathcal{L}_{\mathcal{O}}(x) \leftarrow \mathcal{L}_{\mathcal{O}}(x) \backslash \{u\}$
      end if
    end for
  end for
end procedure

using a set of rules. The rules are applied sequentially from 1a to 7c. While applying these rules, the reduced restrictions may be removed provisionally to avoid using them in the imminent iterations. We are not removing them permanently, as we may need to use such reduced restrictions with the non-reduced ones until we are sure that none of the forthcoming rules may use such a restriction for the reduction anymore. We mark such restrictions as PRs (Provisionally Reduced ones) so that at a later stage we can remove them permanently from the label-set.

Algorithm-1 describes the steps that have to be followed for applying the rules. This algorithm works by taking pairs of restrictions from the label-set and looking for the applicability of the rules. If a rule is applicable, the restrictions will be checked for the following set of conditions to decide whether to resume the refinement or not. The below-mentioned conditions are followed to ensure a quick refinement. The rationales for considering these three conditions are detailed in the longer version of the paper.

- *Condition-1:* No need to further reduce two provisionally reduced (PR) restrictions.
- *Condition-2:* If a rule combines two restrictions ($R1$ and $R2$) and generates either $R1$ or $R2$, then that $R1$ or $R2$ should not be marked as a PR.
- *Condition-3:* If the restrictions of a particular form are *not* used in successive rule-sets, the PR restrictions of such forms can be removed at an early stage.

For illustration, let us consider the label-set of the individual sam. Fig 3 shows the refinement steps and the rules in the rule sets used for the refinement. $\mathcal{L}_{\mathcal{O}}$(sam) is represented vertically. In the figure, the arrows represent the application of rules. The rule numbers are shown in italics. The refinement of two restrictions may sometimes result in more than one restriction. For representing such cases, the arrows are followed by brace brackets ({...}) showing the resultant restrictions.

Initially, the algorithm marks all the labels in the label-set as not PRs. Then the algorithm looks for the applicability of Rule 1a. In the figure, $\mathcal{L}_{\mathcal{O}}$(sam) contains the labels IITStudent and IITPhdStudent whose definitions (in the form of restrictions) are already present in the label-set. Therefore, on applying Rule 1a, they have to be removed from the label-set.

In the algorithm, lines 5-31 consider the rest of the rule-set one at a time and look for possible application of rules on pairs of restrictions in the label-set. In our example label-set, since no rules in the rule sets 2, 3, and 4 were applicable, we move to the next applicable rule set (i.e., Rule-set 5). The algorithm would then apply Rule 5c on two of the restrictions as shown in the figure and refine them to the two restrictions given in the brackets. Application of a rule will be done only if the restrictions in the pair are not marked as PR which is checked using the MARKED_AS_PR method. The *if* condition in line-8 of the algorithm will take care of this. After the application of a rule (using the method APPLY_RULE), the details of the reduced restrictions will be stored in the set variable REF. Based on Condition-2, appropriate changes are made on the contents of REF (lines 14-20). Once all the possible rules in a particular rule set are applied, the reduced restrictions will be marked as PRs (lines 24). Once the algorithm considered all pairs of labels and checked them for the applicability of all the rules in the current rule-set, Condition-3 will be checked for possible permanent removal of the PRs. The entire process will be repeated for all the succeeding rule-sets.

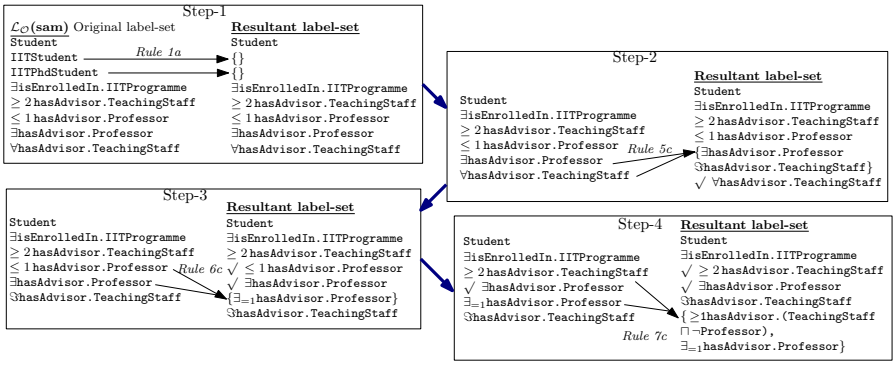


Fig. 3: Steps for the semantic-refinement of $\mathcal{L}_{\mathcal{O}}$(sam). Arrows represent the application of rules.

Coming back to our example label-set, after the application of Rule 5c, one of the reduced restrictions is marked as PR (represented using $\surd$), while the other restriction is not marked as PR due to Condition-2. On changing the rule-set, since no other rules in Rule-set 5 were applicable, the one which is marked as PR can be permanently removed

since Condition-3 is satisfied. In the forthcoming iterations of the for loop (line 5), rules in the rule-set 6 and 7 are applied in a similar fashion. In the last iteration, we will get the most refined set of labels, along with a set of restrictions that are marked as PRs. The restrictions which are marked as PRs are removed to get the refined label-set.

Redundancy score for Label-sets. We introduce the *redundancy-score* to quantify the degree of refinement that we perform on a label-set. Intuitively, this score is intended to capture the amount of redundancy in the NL description that is generated from a label set. This measure is defined in terms of the number of labels in the label set as it plays a role in determining the redundancy and is also based on the refinement rules that we apply while performing the reduction. Initially, the label-set will have a redundancy score of "1" where each label would equally contribute (that is, $1/n$ where n is the number of labels in the label-set) to this score. While applying a rule, the scores (old scores) of the labels that match the antecedents of the rule are redistributed to the new labels (generated as per the consequents of the rule) after multiplying with the weight of the rule. The appropriate weight of the rule is inversely proportional to the rule number as rules are arranged in the increasing order of the amount of redundancy they remove. Therefore, the weight of the rule Rule$_j$ (denoted as w$_j$) is $1/j$. Suppose Rule$_j$ applies to the labels: $\{L_1, ..., L_r\}$, and produces labels: $\{R_1, ..., R_s\}$, then each R_i where $(1 \leq i \leq s)$ is assigned a score as follows. For example, E.g., if *oldScore* of L_1 is 1/8 and that of L_2 is 1/8, then on applying the rule: $L_1 \sqcap L_2 \rightarrow R_1 \sqcap R_2$, the new score of R_1 would be (2/8 * 1/2)*(1/2) = 1/16 and that of R_2 is again 1/16.

$$\mathit{newScore}(R_i) = \frac{w_j \times \sum_{k=1}^{r} \mathit{oldScore}(L_k)}{s} \tag{1}$$

Those label-sets whose redundancy-score remain as "1" even after applying the semantic-refinement algorithm are treated as non-redundant label-sets. Therefore, we have to change the redundancy-score of such label-sets to "0".

Natural Language Descriptions from the Refined Label-sets. In this paper, we have considered a template similar to the following regular expression for generating descriptions of individuals and concepts, (“is”) (“a”) C (“,” | “and”)?)$^+$ (RR (“,” | “and”)?)$^+$

In this regex, C represents the concept name in the label-set, and RR denotes the role restriction in the label-set. The role restrictions are treated in parts. We first tokenize the role names in the constraints. Tokenizing includes word-segmentation and processing of camel-case, underscores, spaces, punctuations, etc. Then, we identify and tag the verbs and nouns in the segmented phase — as R-`verb`, R-`noun` respectively — using NLTK[7]. We then incorporate these segmented words in a *constraint-specific template*, to form a RR. For instance, the restriction $\exists$*hasAdvisor.Professor* is verbalized to “has at least 1 professor as advisor”, using the template: <R-`verb`> at least <*n*><*C*> as <R-`noun`> (where C corresponds to the concept present in the restriction). The constraint-specific templates corresponding to the restrictions are listed in Table-4.

6 Empirical Evaluation

We have done the empirical study to address the following two questions: **Q1:** *Does the semantic-refinement help in improving the understandability of the verbalized knowl-*

[7] Python Natural Language Tool Kit: http://www.nltk.org/

Table 4: Constraint-specific templates of the possible restrictions in a refined label-set.

Restriction	*Constraint-specific template*
$\exists R.C$	<R-verb> at least one $<C>$ as <R-noun>
$\forall R.C$	<R-verb> only $<C>$ as < R-noun >
$\geq nR.C$	<R-verb> at least $<n><C>$ as <R-noun>
$\leq mR.C$	<R-verb> at most $<m><C>$ as <R-noun>
$\Im R.C$	<R-verb> at least one $<C>$ & only $<C>$ as <R-noun>
$\exists_{=n} R.C$	<R-verb> exactly $<n><C>$ as <R-noun>

edge? **Q2:** *Is the semantic refinement helpful in validating the correctness of ontology axioms?* For answering these questions, we present the domain experts with two representations of the same knowledge: one is from the label-sets having redundancy score "1", and the other from the refined label-sets (that is, with redundancy score < 1). We call the former as the ones from the *baseline approach* and the latter as those from the *proposed approach*. The descriptions generated using the baseline approach are similar to the texts generated using an existing ontology verbalizer. Table 5 shows the examples of the descriptions generated using both approaches.

Table 5: Examples of the descriptions of individuals and concepts from PD, HarryPotter (HP) and Geographical Entity (GEO) ontologies, generated using the proposed and baseline approaches

Proposed approach	*Baseline approach* (with redundancy score =1)	*Ontology*
`Bird cherry Oat Aphid`: is a biotic-disorder, having at least one pest-insect and all its factors are pest-insects. (Redundancy score = 0.340)	`Bird cherry Oat Aphid`: is a disorder, bio-disorder, pest damage and insect damage. It is all the following: has as factor only pest-insect, has as factor only pest, has as factor only organism and has as factor something.	PD
`Mite Damage`: is a pest damage, having at least one mite pest and all its factors are mite pests. (Redundancy score = 0.324)	`Mite Damage`: is a disorder, a biotic-disorder and a pest damage. It is all the following: has as factor only organism, has as factor only pest, has as factor only mite pest, has as factor at least one thing.	PD
`Hermione Granger`: is a Hogwarts Student, a muggle, a gryffindor, having exactly one cat as pet. (Red. score = 0.425)	`Hermione Granger`: is a Hogwarts student, a student, a human, a muggle, a gryffindor. It is all the following: has a pet, has as pet a cat, has as pet only creature, has at least one creature, has at most one creature, as pet.	HP
`Hogwarts Student`: is a Student, is a Gryffindor or Hufflepuff or Ravenclaw or Slytherin, and having exactly one pet. (Redundancy score = 0.350)	`Hogwarts Student`: is a student, a human, is a Gryffindor or Hufflepuff or Ravenclaw or Slytherin. It is all the following: has a pet, has as pet only creatures, has at least one creature, has at most one creature.	HP
`Aggregate of sovereign states`: is not a gov. organization, is aggregate of only sovereign states and is aggregate of at least two sovereign states. (Red. score = 0.324)	`Aggregate of sovereign states`: is not a gov. organization and not a sovereign state. It is all the following: is aggregate of only governmental organization, is aggregate of at least two governmental organizations, is aggregate of only sovereign states and aggregate of at least two sovereign states.	GEO
`Florida`: is a gov. organization and a major administrative subdivision, is related to at least one nation as a part, is related to exactly one sovereign state as a member, and is a subordinate authority of at least one sovereign state. (Red. score = 0.204)	`Florida`: is a major administrative subdivision, an organization, a gov. organization, a subnational entity. It is all the following: is a part of at least one nation, is a subordinate authority of at least one sovereign state, is a member of at least one sovereign state and have at most one member of relationship with sovereign state.	GEO

For Q1, the domain experts were asked to rate the *degree of understanding* of the descriptions in the scale: (a) *Poor*; (b) *Medium*; (c) *Good*. And, for Q2, to measure the *usefulness* of the generated descriptions for validating the domain knowledge, the domain experts were told to choose one from the options: (a) *Valid* (b) *Invalid* (c) *Don't know* (d) *Cannot be determined*. If they cannot distinguish a given sentence to be "Valid" or "Invalid" because of their lack of knowledge, then they are instructed to choose the

third option "Don't Know". Option (d) is to be selected if the expert finds it difficult to reach a conclusion on the validity of the sentence–which means, the description is either ambiguous or confusing. We have used two online available ontologies for generating descriptions: (1) *Plant Disease (PD)* ontology, and (2) *Data structures and Algorithms (DSA)* ontology. These ontologies can be downloaded from our website[8]. The PD ontology has 546 individuals, 105 concepts, and 15 object properties, and the DSA ontology has 333 individuals, 53 concepts, 19 object properties, and 11 datatype properties.

Experimental setup. After generating descriptions from the aforementioned ontologies, since the manual evaluation of all the generated descriptions is difficult, a small number of descriptions were utilized for the study. We have selected a representative set (and a heterogeneous set) of descriptions by grouping all the descriptions based on their label-sets and then randomly choosing one description from each group. From PD ontology, 31 descriptions of individuals and 10 descriptions of concepts were considered for evaluation. Similarly, from DSA ontology, 14 descriptions of individuals and 17 descriptions of concepts were chosen. Then, experts from the two domains were asked to review the verbalized descriptions. To avoid bias, the reviewers were not informed about the approach followed for generating the description, and the descriptions were randomly presented via a google form. In addition, to finding the inter-rater agreement among the experts, we have also recorded the confidence score of each reviewer for a given question such that in the case of a conflict we make a decision based on their scores. Seven experts from the PD domain and fourteen experts of DSA were involved in the study.

6.1 Results and Discussions

Fig 4-7 show the summary of the ratings given by the domain experts.

Q1: The degree of understanding of a description is identified by examining the ratings (i.e., poor, medium, or good) given by the domain experts. The domain experts were asked to choose 'poor'or 'medium'as the level of understanding if there is any ambiguity in the description. To confine the reasons for ambiguity to the fidelity to OWL constructs alone, possible (manual) grammatical error corrections have been done on the generated text—as we were not using any sophisticated NL generation techniques. Grammatical errors such as subject-verb agreement errors, verb tense errors, verb form errors, singular/plural noun ending errors, and sentence structure errors were corrected uniformly (and in an unbiased way) for both the approaches. Fig 4 and Fig 5 show the summary of the responses (in percentage) which we received for the descriptions of PD ontology and for the descriptions of DSA ontology, respectively. In both cases, since the Fleiss' kappa scores were in the *substantial agreement* range, the overall ratings are calculated by considering the majority responses. For PD ontology, 32 out of the 41 descriptions generated using the proposed approach were rated as 'good', whereas, for those generated using the baseline approach, only 6 out of 41 texts were rated as 'good'. For DSA ontology, 23 out of 31 descriptions generated by the proposed approach were 'good', only 12 descriptions generated using the baseline approach were

[8] https://sites.google.com/site/ontoworks/ontologies (all ontologies used are available here)

rated as ‘good’. These results highlight the significance of the semantic-refinement process in domain knowledge understanding.

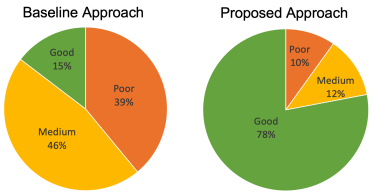


Fig. 4: Summary of the ratings obtained for the descriptions from the **PD** ontology

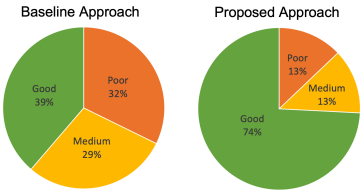


Fig. 5: Summary of the ratings obtained for the descriptions from the **DSA** ontology

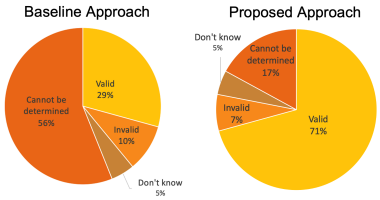


Fig. 6: Summary to determine the usefulness of the generated descriptions in validating the **PD** ontology

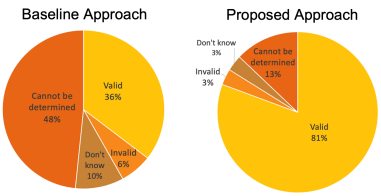


Fig. 7: Summary to determine the usefulness of the generated descriptions in validating the **DSA** ontology

Q2: Fig 6 and 7 show the statistics to determine the usefulness of the generated descriptions in validating the correctness of two domain ontologies. Usefulness of the generated descriptions in validating the correctness of an ontology is obtained by looking at the number of descriptions which are marked as ‘Cannot be determined’. The three options: ‘Valid’, ‘Invalid’and ‘Don’t know’, imply that the text is useful in getting into a conclusion, whereas the option ‘Cannot be determined’ indicates that there is some problem in the representation. From Fig 6 and Fig 7, in case of the proposed approach, only 7 out of 41 descriptions from PD ontology and 4 out of 31 descriptions from DSA ontology were not useful in determining the quality of the ontology, whereas in case of the baseline approach, approximately 50 percentage of the descriptions were not helpful. This clearly indicates that, verbalization after semantic-refinement is highly effective in applications such as ontology validation.

Discussion and future work. In this paper, we have formally defined the notion of redundancies in a label-set and a technique to systematically reduce the redundancies. However, the notion of redundancy is, to some extent, subjective. That is, depending on the readers’ domain knowledge, the level of redundancy in the text varies. In the process of semantic-refinement, we remove the generic information from the label-set with an assumption that the human readers would be familiar with the explicit relationships

between the domain entities. In that sense, a reader with poor domain knowledge may miss out on generic concept information due to the refinement process. This would be easily visible when the concept hierarchies are reduced to the specific ones alone. One possible way to overcome this problem is by including relevant (but, not all) concept names, that were previously omitted in the semantic-refinement process, in the refined label-set. E.g., in Table 5, we can further generalize the description of the concept `mite damage`, by including additional generic concept details, as "*Mite Damage is a pest damage and a biotic-disorder, having at least one mite pest and all its factors are mite pests.*" Since only a generic concept name is included in addition to all the refined concepts, the meaning of the description is not affected. More investigation and empirical studies related to this could be done as a future endeavor. Another interesting method (which is not addressed in this paper) to improve the description of individuals is by considering the property assertions along with the label-sets while generating descriptions. Considering property relationships/assertions is important because validation of an ontology also involves verifying the truthfulness of the property assertions in it.

7 Conclusion

A novel approach for generating natural language descriptions of ontology entities is presented in the paper. The generated descriptions were not merely verbatim translations of logical axioms of the ontology. Instead, they were generated from a refined set of logical restrictions satisfied by individuals/concepts under consideration. We have proposed seven sets of refinement rules and an algorithm for this refinement process. We have observed that the proposed method indeed gives us short, precise, and comprehensive descriptions of individuals and concepts. Our time-budgeted empirical studies based on two ontologies have shown that the redundancy-free description of the domain knowledge is helpful in understanding the formalized knowledge more effectively and is also useful for validating them, typically for the humans who are experts of the domain under consideration.

Acknowledgements

This project is funded by Ministry of Human Resource Development, Gov. of India. We express our fullest gratitude to the participants of our evaluation process: Dr. S.Gnanasambadan (Director of Plant Protection, Quarantine & Storage), Ministry of Agriculture, Gov. of India; Mr. J. Delince and Mr. J. M. Samraj, Department of Social Sciences AC & RI, Killikulam, Tamil Nadu, India; Ms. Deepthi.S (Deputy Manager), Vegetable and Fruit Promotion Council Keralam (VFPCK), Kerala, India; Dr. K.Sreekumar (Professor) and students, College of Agriculture, Vellayani, Trivandrum, Kerala, India. We also thank all the undergraduate and post-graduate students of Indian Institute of Technology, Madras, who have participated in the empirical study.

Bibliography

Harith Alani, Stephen Harris, and Ben O'Neil. Winnowing ontologies based on application use. In *The Semantic Web: Research and Applications*, pages 185–199, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-34545-9.

Ion Androutsopoulos, Gerasimos Lampouras, and Dimitrios Galanis. Generating natural language descriptions from OWL ontologies: the naturalowl system. *CoRR*, abs/1405.6164, 2014. URL http://arxiv.org/abs/1405.6164.

Anne Cregan, Rolf Schwitter, and Thomas Meyer. Sydney owl syntax - towards a controlled natural language syntax for owl 1.1. In Christine Golbreich, Aditya Kalyanpur, and Bijan Parsia, editors, *OWLED*, volume 258, 2007.

Kathrin Dentler and Ronald Cornet. Intra-axiom redundancies in snomed ct. *Artif. Intell. Medicine*, 65(1):29–34, 2015. URL http://dblp.uni-trier.de/db/journals/artmed/artmed65.html#DentlerC15.

Vinu E. Venugopal and P. Sreenivasa Kumar. Difficulty-level Modeling of Ontology-based Factual Questions. *Semantic Web*, 11(6):1023–1036, 2020. https://doi.org/10.3233/SW-200381.

Stephan Grimm and Jens Wissmann. Elimination of redundancy in ontologies. In *Proceedings of the 8th ESWC on The Semantic Web: Research and Applications*, pages 260–274. Springer-Verlag, 2011. ISBN 978-3-642-21033-4.

Glen Hart, Catherine Dolbear, and John Goodwin. Lege feliciter: Using structured english to represent a topographic hydrology ontology. In *OWLED*, volume 258 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

Daniel Hewlett, Aditya Kalyanpur, Vladimir Kolovski, and Christian Halaschek-wiener. Effective nl paraphrasing of ontologies on the semantic web. In *End User Semantic Web Interaction Workshop (ISWC 2015)*, 2005.

Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Reasoning with individuals for the description logic shiq. *CoRR*, cs.LO/0005017, 2000.

Mustafa Jarrar, C. Maria, and Keet Paolo Dongilli. Multilingual verbalization of orm conceptual models and axiomatized ontologies. Technical report, 2006.

Kaarel Kaljurand and Norbert E Fuchs. Verbalizing owl in attempto controlled english. In *OWLED*, volume 258, 2007.

Paolo Liberatore. Redundancy in logic I: CNF propositional formulae. *AI*, 163(2): 203–232, 2005. https://doi.org/10.1016/j.artint.2004.11.002.

Silvio Peroni, Enrico Motta, and Mathieu d'Aquin. Identifying key concepts in an ontology through the integration of cognitive principles with statistical and topological measures. In *The Semantic Web*, volume 5367, pages 242–256. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-89703-3.

Md Kamruzzaman Sarker, Joshua Schwartz, Pascal Hitzler, Lu Zhou, Srikanth Nadella, Brandon Minnery, Ion Juvina, Michael L. Raymer, and William R. Aue. Wikipedia Knowledge Graph for Explainable AI. In *Knowledge Graphs and Semantic Web Conference (KGSWC)*, 11/2020 2020.

M. Schmidt-Schau and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

Robert Stevens, James Malone, Sandra Williams, Richard Power, and Allan Third. Automating generation of textual class definitions from owl to english. *J. Biomedical Semantics*, 2(S-2):S5, 2011.

Allan Third. Hidden Semantics: What can we learn from the names in an ontology? In *Proceedings of the Seventh International Natural Language Generation Conference*, pages 67–75, Stroudsburg, PA, USA, 2012. ACL.

Allan Third, Sandra Williams, and Richard Power. OWL to English: a tool for generating organised easily-navigated hypertexts from ontologies. In *10th International Semantic Web Conference (ISWC 2011)*, October 2011.

Simple and Fast Methods for Integrating Predicted Data into Bayesian Optimization*

Simona Capponi[1], Andy I. Cooper[2], John Fearnley[1], and Vladimir Gusev[2]

[1] Department of Computer Science, University of Liverpool, UK
[2] Department of Chemistry, University of Liverpool, UK

Abstract. We propose two simple and fast methods to optimize a black-box function while exploiting an inexpensive predictive model, which generates a deterministic predicted value at any input point in the search space. In contrast to prior work on multi-fidelity optimization, our setup assumes that there is only one predictor whose accuracy level is unknown. We also assume that querying the predictor is essentially free compared to the actual objective function, thus no cost is assigned to it. We show that our methods generally outperform the existing multi-fidelity approaches for this scenario, while requiring remarkably less computational time.

Keywords: Bayesian optimization · Predicted data · Multi-fidelity optimization.

1 Introduction

In this paper, we study a problem that is motivated by the design of experiments that are based on Bayesian optimization. In this setting, there is a range of input parameters, which could be various processing conditions or ratios of input chemicals, and there is a desired objective, such as maximizing some desired property of the resulting product or improving its yield.

Bayesian optimization has emerged recently as one of the leading methods to address this type of problems across natural sciences [3, 8, 12, 28] expanding beyond its traditional applications [6, 27, 32]. It considers a function $f : \mathcal{X} \to \mathbb{R}$ that maps some input domain $\mathcal{X}$ to objective values. At each step of the optimization, the method selects some point $x \in \mathcal{X}$ and requests the value of $f(x)$. Once this value is provided, the process is repeated.

The problem with using this approach for real-world scientific experiments is that in this setting, evaluating f is a very costly operation. For example, it could involve using very expensive input chemicals, it could require many hours of work from a human to set up the experiment, and the results may take days or weeks to arrive. Therefore it is crucial that the optimization process makes as few queries to f as possible.

On the other hand, real-world scientists do not treat their experiments like black-box functions that need to be optimized. There is often a wealth of domain

* This work was supported by The Leverhulme Trust

knowledge available such as theoretical results or simulators that are able to predict the outcome of any particular experiment. However, these predictions can be fairly coarse, with no formal guarantee on their accuracy, and they may give completely incorrect answers for some portions of the domain.

In this paper, we encode this as a *predictor function* $p : \mathcal{X} \to \mathbb{R}$, which makes a prediction about the value of f for each point in the domain. We assume that f is expensive to evaluate, but that p can be evaluated essentially for free. We seek to use the potentially low quality predictions made by p to accelerate the optimization of f.

In this work we assume that the predictor is deterministic, so querying the same point multiple times yields no extra information. The predictor may also make systemic errors: if the predictor makes a poor quality prediction for a particular point x, then it may well also make poor quality predictions for the points surrounding x. So, if one is in a region where the predictor gives poor predictions, then there may be no easy way to extract information about f, and so the main challenge is to discard these poor quality predictions while making use of the good quality predictions where they arise.

Our contribution. We propose two simple methods for integrating the predictor into Bayesian optimization. The first, which we call the *exclusion radius* method, adds predicted data at the start of the optimization process, and then iteratively deletes it as real data is obtained. The second, which we call *discrepancy prediction*, also adds predicted data at the start of the process, and then as more information about f is obtained, it attempts to correct the errors in the predicted data by learning a model of the difference between p and f.

We present experimental results for each of these methods. We test the methods on standard benchmarks for Bayesian optimization, and our goal was to test the methods on predictors of varying accuracies, where the accuracy of a predictor is defined as the mean squared error between p and f (see Equation (3)). Thus, predictors that make larger errors on average are less accurate.

To carry out these experiments, we needed to to build predictors at a specified accuracy level for a given benchmark function. In Section 3 we present a method to build a deterministic smooth predictor for a given function and a given accuracy level. Figure 1 shows the results of this method the Michalewicz benchmark function, where it can be seen that the predictors give information about f, but with local errors that increase as the error level increases.

We then benchmark our methods against standard Bayesian optimization (which ignores the predictor), and against a standard multi-fidelity approach that uses the predictor as a lower-fidelity model. Our results show that the exclusion radius method is competitive in all scenarios, while the discrepancy prediction method is less consistent, but it can work well for benchmarks with lower levels of error. Our results also show that our methods have a particular advantage early on in the optimization process, where in many cases they are able to quickly find points that have reasonably low regret. We also find that our methods are significantly faster than the multi-fidelity approach.

Related work. Prior work has considered methods for warm-starting [17,26,31] Bayesian optimization. The warm starting approach uses results from optimization tasks which have already been solved on previous datasets to introduce information into the optimization process, and therefore potentially exploiting similarities between the previous datasets and the current one. For example Feurer et al. and Kim et al. [4, 17] initialize Bayesian optimization by fitting the initial surrogate model with optimum solutions found for the previous tasks, before running the current task. Alternatively the objective functions learned in the previous tasks, f_{old}, can be exploited to infer information about the current objective, f, by iteratively learning the difference between f and f_{old} [4,17]. This can be contrasted with our scenario, where we have no prior knowledge about f, and no prior runs on related optimization problems, but we instead have the potentially poor quality predictions made by p.

Multi-fidelity optimization methods have been developed to deal with scenarios where the optimization process has access to lower fidelity models which approximate the actual objective function, and which can be evaluated at reduced cost [5, 11, 15, 16, 24, 25, 29]. In this setting it is often assumed that these models are hierarchically ordered by their fidelity with respect to the actual objective, such that as one moves up the hierarchy the cost of evaluating the models decreases, but with the drawback of obtaining lower fidelity information about the function that is being optimized. The main goal of multi-fidelity optimization is to reach an optimal trade-off between cost and fidelity, thus minimizing the overall cost. The fidelity level of the model to sample and the next input point where to evaluate the model are selected simultaneously by maximizing specifically designed acquisition functions [25, 33].

The scenario we consider in this paper is slightly different, but related. As our motivation arises from experimental work in natural sciences, and due to the extremely high costs involved in those experiments, we assume that the cost of evaluating f is extremely large compared to the cost of evaluating p, meaning that obtaining predicted data is essentially free. So whereas existing work in multi-fidelity optimization often carefully balances the costs of obtaining predicted data as opposed to obtaining real data [18, 25], in our set up this is meaningless, as there is essentially no cost to obtaining as much predicted data as is needed at the start of the optimization process. We also make no formal assumptions about the quality of the data produced by p, because in our setting it does not make sense to do so. While a scientist may have a theory that predicts the outcome of an experiment, the only way to validate the accuracy of that theory would be to run experiments. But that would be self-defeating, as it is those expensive experiments that we would like to avoid in the first place.

2 Our Methods

In this paper we study an optimization problem defined by a continuous domain $\mathcal{X}$ and a black-box function $f : \mathcal{X} \to \mathbb{R}$ over that domain, where the goal is to find a point $x \in \mathcal{X}$ that maximizes or minimizes f.

Algorithm 1: Standard Bayesian Optimization

```
initialize D_0
for n ← 0, 1, ... do
    Update the statistical model M
    Select x_{n+1} by optimizing the acquisition function a:
    x_{n+1} = arg max_{x∈X} a(x|D_n, M)
    Query the objective function f to obtain y_{n+1}
    Augment data D_{n+1} = {D_n, (x_{n+1}, y_{n+1})}
    if stopping condition is reached then
        break
    end
end
```

Bayesian optimization. Bayesian optimization (BO) is a global optimization method developed to address black-box optimization problems, and is shown in Algorithm 1. BO models f with a probabilistic surrogate model, $\mathcal{M}$, which describes the probability distribution over all possible functions, conditioned over a training data set.

The most common choice for $\mathcal{M}$ is a Gaussian process (GP) [32], which assumes that f follows a multivariate Gaussian distribution:

$$f(x) \sim \mathcal{N}(m(x), k(x, x')), \tag{1}$$

where $x \in \mathcal{X}$ is an input point in the search space, $m(x)$ is the expected value of the objective function, i.e. $m(x) = \mathbb{E}[f(x)]$, while $k(x, x')$ is the covariance function, and it represents the uncertainty over the estimation of Equation (1).

The posterior distribution of the GP is used to calculate the acquisition function $a(x)$, an inexpensive utility function which is maximized in order to find the best candidate $x^* \in \mathcal{X}$ to sample. Then, the new data $\{(x^*, f(x^*)\}$ is added into the training set to improve the accuracy of the surrogate model.

Common choices for the acquisition function are expected improvement [14, 21], upper confidence bound, or entropy search [9]. In our methods we use expected improvement, which is defined as: $EI(x) = \mathbb{E}\left[max\left(f^{best} - f(x), 0\right)\right]$, where f^{best} is the best value of f found so far. In other words the expected improvement measures how much progress we expect to make towards the actual optimum by evaluating the objective function at a point x.

Predictors. In this paper, we assume that the true objective function $f : \mathcal{X} \to \mathbb{R}$ is expensive to evaluate, but comes equipped with a potentially low quality but inexpensive predictive model $p : \mathcal{X} \to \mathbb{R}$, which we will call the *predictor*. We do not make any formal assumption on the predictor nor on the process in which it is generated: it may be implemented using a theoretical model, through computational simulations, or it could be a machine learning model.

The quality of the predictor at any given point can be quantified by the *discrepancy*, $\delta : \mathcal{X} \to \mathbb{R}$, which we define as

$$\delta(x) = f(x) - p(x). \tag{2}$$

Our goal is to use the outputs generated by the predictor to speed up the process of Bayesian optimization.

Algorithm 2: The exclusion radius method.

Input: The initial exclusion radius r, a predictor p, and a number of real observations N_{obs}

Initialize the starting sets of real and predicted points respectively as $\mathcal{R}_0 = \emptyset$, $\mathcal{P}_0 = \{(x_j, p(x_j)\}_{j=1,\cdots,N}$ and set the initial data-set $\mathcal{D}_0 = \mathcal{R}_0 \cup \mathcal{P}_0$
for $i \leftarrow 0, 1, ...$ **do**
  Update the surrogate model $\mathcal{M}$ using the whole data-set $\mathcal{D}_i$.
  Select a new point $\mathbf{x}_{i+1}$ by optimizing the acquisition function a:
    $\mathbf{x}_{i+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} a(\mathbf{x}, \mathcal{M})$
  Query f to obtain $\mathbf{y}_{i+1} = f(\mathbf{x}_{i+1})$
  Create a new real data set $\mathcal{R}_{i+1} = \mathcal{R}_i \cup \{(\mathbf{x}_{i+1}, \mathbf{y}_{i+1})\}$
  Find the predicted points in a ball around $\mathbf{x}_{i+1}$:
    $\mathcal{B}_{i+1} = \{(x, y) \in \mathcal{P}_i : \|x - x_{i+1}\|_2 \leq r\}$
  Exclude those points by setting $\mathcal{P}_{i+1} = \mathcal{P}_i \setminus \mathcal{B}_{i+1}$
  Set $\mathcal{D}_{i+1} = \mathcal{R}_{i+1} \cup \mathcal{P}_{i+1}$
  if *number real observations* $= N_{obs}$ **then**
    break
  end
end

The exclusion radius method. Our first method is called the *exclusion radius method*, and it is shown in Algorithm 2. The idea is to sample a large number of points from the predictor and use these to initialize Bayesian optimization. Then we run Bayesian optimization as normal, but in each iteration, when $f(x)$ is sampled at a point $x \in \mathcal{X}$, we remove all predicted points that are within a given radius of x from the model.

There are two main advantages of this approach. Firstly, by initializing the model with predicted points, we give the model a warm-start, and thus our initial queries to f will be informed by the data from the predictor. Secondly, by removing points that are close to real data, we are able to discard potentially inaccurate predicted data when more accurate data from f has been obtained.

Formally, in each iteration i, the method maintains two sets of data. The set $\mathcal{R}_i$ denotes the set of *real* data, and it contains pairs $(x, f(x))$. The set $\mathcal{P}_i$ denotes the set of *predicted* data, and it contains pairs $(x, p(x))$. At the start of the process $\mathcal{R}_0$ is empty, since we have not made any queries to f, and $\mathcal{P}_0$ is initialized with a set of initial predicted points, which we will choose randomly in our experiments. In each step, Bayesian optimization proposes a new point $x_i \in \mathcal{X}$ to be sampled. The new data $(x_i, f(x_i))$ is added to $\mathcal{R}_i$, and then all predicted points that are close to x_i are deleted from $\mathcal{P}_i$. Specifically, all predicted points within distance r from x_i are deleted from $\mathcal{P}_i$. Thus, the set of points to

delete is given by $\mathcal{B} = \{(x, y) \in \mathcal{P}_0 : \|x - x_i\|_2 \leq r\}$. The radius r is a parameter of the algorithm.

Algorithm 3: The discrepancy prediction method.

Input: predictor p, number of real observations N_{obs}
Initialize the $\mathcal{R}_0$, $\mathcal{P}_0$, and $\mathcal{C}_0$ as mentined in the text, and set $\mathcal{S}_0 = \mathcal{R}_0 \cup \mathcal{P}_0$
for $i \leftarrow 0, 1, ...$ **do**
  Update the surrogate model $\mathcal{M}$ using the data in $\mathcal{S}_i$.
  Select a new point $\mathbf{x}_{i+1}$ by optimizing the acquisition function a:
  $\mathbf{x}_{i+1} = \arg\max_{\mathbf{x} \in \mathcal{D}} a(\mathbf{x}, \mathcal{M})$
  Query f to obtain $\mathbf{y}_{i+1} = f(\mathbf{x}_{i+1})$
  Create a new real data set $\mathcal{R}_{i+1} = \mathcal{R}_i \cup \{(\mathbf{x}_{i+1}, \mathbf{y}_{i+1})\}$
  Set $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{(\mathbf{x}_{i+1}, f(\mathbf{x_{i+1}}) - p(\mathbf{x_{i+1}}))\}$, and then retrain $\mathcal{M}_\delta$ on $\mathcal{C}_{i+1}$
  Create $\mathcal{P}_{i+1} = \{(\mathbf{x}, p(\mathbf{x}) + \hat{\delta}(\mathbf{x})) \ : \ \mathbf{x} \text{ is a point in } \mathcal{P}_0\}$
  Set $\mathcal{S}_{i+1} = \mathcal{R}_{i+1} \cup \mathcal{P}_{i+1}$
  if *number real observations* $= N_{obs}$ **then**
    break
  end
end

Discrepancy prediction. Our second method is called the *discrepancy prediction method*, and is shown in Algorithm 3. This method maintains an estimation of the discrepancy of a point, as defined in Equation (2). This will be modelled by a Gaussian process that will be trained during the course of the Bayesian optimization. Hence, this method uses two Gaussian processes: the model $\mathcal{M}$ that is used as part of Bayesian optimization, and a model $\mathcal{M}_\delta$ that is used to predict the discrepancy.

As we proceed with Bayesian optimization, we will maintain a set $\mathcal{C}_i$ which will contain data on the discrepancy of all points that we have sampled from f. That is, every time we sample $f(x)$ we add $(x, f(x) - p(x))$ to $\mathcal{C}_i$. Then we train $\mathcal{M}_\delta$ using $\mathcal{C}_i$, and we define $\hat{\delta} : \mathcal{D} \to \mathbb{R}$ to be the expected value $\delta(x)$ of each point $x \in \mathcal{D}$ as predicted by $\mathcal{M}_\delta$.

Like the exclusion radius method, we split the data into real points $\mathcal{R}_i$, and predicted points $\mathcal{P}_i$. Unlike that method, we will not delete any predicted points. Instead, in each iteration we update the values using the new discrepancy prediction. Formally, to do this update we create the set $\mathcal{P}_i$ so that it contains $(x, p(x) + \hat{\delta}(x))$ for each predicted point x, where $\hat{\delta}(x_p)$ is the expected discrepancy, inferred from the values of $\hat{\delta}$ given by $\mathcal{M}_\delta$.

We initialize this method using a mix of predicted points and real points (with the split being 45 predicted points and 5 real points in our experiments). This allows us to train an initial discrepancy predictor, and then make an initial adjustment of the predicted data before the optimization process begins. So

formally, if we initialize with k real points and l predicted points, then we set $\mathcal{R}_0 = \{(x_i, f(x_i)) \ : \ i = 1, 2, \ldots, k\}$ where each x_i is chosen uniformly from the domain $\mathcal{D}$, and we initialize $\mathcal{C}_0 = \{(x_i, f(x_i) - p(x_i)) \ : \ i = 1, 2, \ldots, k\}$ for those same points. Then we train M_δ on $\mathcal{C}_0$, and we set $\mathcal{P}_0 = \{(x_i, p(x) + \hat{\delta}(x)) \ : \ i = 1, 2, \ldots, l\}$ as the initial set of predicted points.

3 Creating Predictors

The functions. In our experimental results, we will test our methods on five standard benchmark functions: Ackley, Griewank, Michalewicz [13], Rastrigin [1] and Styblinski-Tang [30]. The analytical form of these functions as well as their global minima and the search domain over which they are optimized are summarized in Table 1.

Name	Formula	Minimum	Search domain
Ackley	$-20\exp\left[-0.2\sqrt{0.5\sum_{i=1}^{2} x_i^2}\right] - \exp\left[0.5\sum_{i=1}^{2}\cos(2\pi x_i)\right] + e + 20$	$f(0,0) = 0$	$-4 \leq x_i \leq 4$
Griewank	$1 + \frac{1}{4000}\sum_{i=1}^{2} x_i^2 - \prod_{i=1}^{2}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	$f(0,0) = 0$	$-10 \leq x_i \leq 10$
Michalewicz	$-\sum_{i=1}^{2}\sin(x_i)\sin^{20}\left(\frac{i\,x_i^2}{\pi}\right)$	$f(2.20, 1.57) = -1.801$	$0 \leq x_i \leq \pi$
Rastrigin	$20 + \sum_{i=1}^{2}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	$f(0,0) = 0$	$-5.12 \leq x_i \leq 5.12$
Styblinski-Tang	$\frac{1}{2}\sum_{i=1}^{2}\left(x_i^4 - 16x_i^2 + 5x_i\right)$	$f(-2.0935, -2.0935) \simeq -78.33$	$-5 \leq x_i \leq 5$

Table 1. The benchmark functions that we use

While these benchmarks are standard, they do not come with any pre-defined predictor functions, so in order to benchmark our methods we must build the predictors ourselves. In particular, we will build predictors with the following properties:

- The predictor will be deterministic.
- The predictor will be smooth.
- The accuracy of the predictor will be proportional to a given error parameter N, which will allow us to build predictors at any given accuracy level.

To achieve this we will use the following high-level procedure.

1. Select a number of points from the domain.
2. Assign each point an offset of $f(x) + N$ or $f(x) - N$ uniformly at random.

3. Use these points and values to train a Gaussian process over the entire domain.
4. Use the posterior mean of the GP as the predictor.

In this way we obtain a predictor that is a deterministic smooth function over the entire domain, and whose predictions are influenced by the $+N$ and $-N$ values at the chosen points. Note that this does not imply that *all* points get predicted values of $f(x) + N$ or $f(x) - N$: these values are simply the training inputs to the GP, and it is the posterior mean of that GP that gives the actual predicted values.

In the rest of the section we describe this procedure in full detail.

Creating the predictors. We begin by selecting a set of points from the space $\mathcal{A} \subseteq \mathcal{X}$, which will be the set of points used to train the GP. The set $\mathcal{A}$ is chosen according to a Latin hypercube sampling [19] that is overlaid on the space. Latin hypercube sampling is a method for generating a near-random samples from a multivariate distribution. Compared to random sampling, it is able to reduce the number of samples necessary to approach the real distribution of the sampled function [20]. We then randomly perturb each point in $\mathcal{A}$, which further reduces the regularity of the point set. Using Latin hypercube sampling to generate the initializing points introduces randomness in the sampling of the input points, while still covering the whole search space. In full detail, our technique is as follows.

1. First we generate an initial set of 400 input points $\mathcal{A} \subset \mathcal{X}'$, according to a Latin hypercube sampling [19] where $\mathcal{X}' \supset \mathcal{X}$ is obtained by extending the original search space by 10% in each dimension.
2. To further increase the variability between different predictors we create a second set of points $\mathcal{A}'$ in which each point in $\mathcal{A}$ is translated by a random offset. To do this we take each point $(x, y) \in \mathcal{A}$ and we construct the point $(x + \alpha,\ y + \beta)$, where α and β are distributed according to $\mathcal{N}(0, 0.2)$.
3. Each point $x \in \mathcal{A}'$ is randomly assigned a value $v(x)$ of either $f(x) + N$ or $f(x) - N$ with the probability of either choice being 0.5.
4. The set of points $\{(x, v(x) : x \in \mathcal{A}'\}$ are fitted using a Gaussian process with squared exponential kernel, $GP\,(m(x), k(x, x'))$. The posterior mean of the GP, $m(x)$, is our predictor p which returns an expected value $p(x)$ for each point $x \in \mathcal{X}$.

In Step 2, we extend the domain beyond the original search space to ensure that the predictor gives reasonable answers on the boundary of the domain. Without this, the GP will have high variance on the boundary, leading to poorer quality predictions on the boundary relative to the rest of the space.

Creating predictors for benchmarking. The method that we have just outlined generates a wide variety of predictors, but these predictors have a large range of accuracies even when the error parameter N is fixed. Moreover, different benchmark functions react differently to increases in the error parameter: some

see a large increase in the error of the predictor, while some see a smaller increase. The end result is that we cannot use N itself to define a error level that holds across all benchmarks.

We will address this by building sets of predictors that have a particular level of error. We use the following as a measure of accuracy

$$\hat{E} = \frac{\sqrt{\sum_{i=1}^{M} [f(x_i) - p(x_i)]^2}}{M}, \tag{3}$$

which is the mean squared error between f and p averaged over a set of M points. For our setup, we sampled $\hat{E}$ according to a grid containing $M = 30000$ points. However, $\hat{E}$ is not normalized across different benchmark functions. To address this we define Δf to be the difference between the maximum and minimum values of f: $\Delta f = \max_{x \in \mathcal{X}} [f(x)] - \min_{x \in \mathcal{X}} [f(x)]$, and we define the *accuracy* of a predictor to be $\text{acc}(p) = \hat{E}/\Delta f$.

We then fix three target error levels for our benchmarks: we choose accuracies of 0.05, 0.10, and 0.15, which we refer to as low, medium, and high error, respectively. To generate predictors with these accuracies, for each benchmark function we performed a binary search over values of N: for each value we generated 100 predictors, and then adjusted upwards or downwards depending on whether the average accuracy was too high or to low relative to the target accuracy. Then, once an appropriate value of N was found, there was still a wide range of accuracies in the generated predictors, so we excluded all predictors that were further than 5% of the target accuracy. We ensured that, in all cases, there were at least 20 generated predictors left in the benchmarking set. The resulting predictors and error levels are shown in Figure 1.

4 Experimental Setup

In this section we describe our overall experimental setup, and the setup for each of the methods. A summary of our parameters can be found in Tables 2 and 3 in Appendix A.

We benchmark our methods against the standard Bayesian optimization method that does not use the predictor at all, and against a standard multi-fidelity approach that treats the predictor as a lower fidelity model.

For a fair comparison we run both our methods and multi-fidelity until the objective function f was queried a fixed number of times, so that all the methods could exploit an equal amount of real data. In all experiments, we set this number to 80 real queries.

For standard Bayesian optimization, and our two methods, this means that the number of steps is fixed, since those methods query one real point in each iteration, while for the multi-fidelity approach the number of iterations was unbounded, since any iteration that queried the predictor was not counted, though in practice we stopped the method after 1000 total iterations had been completed.

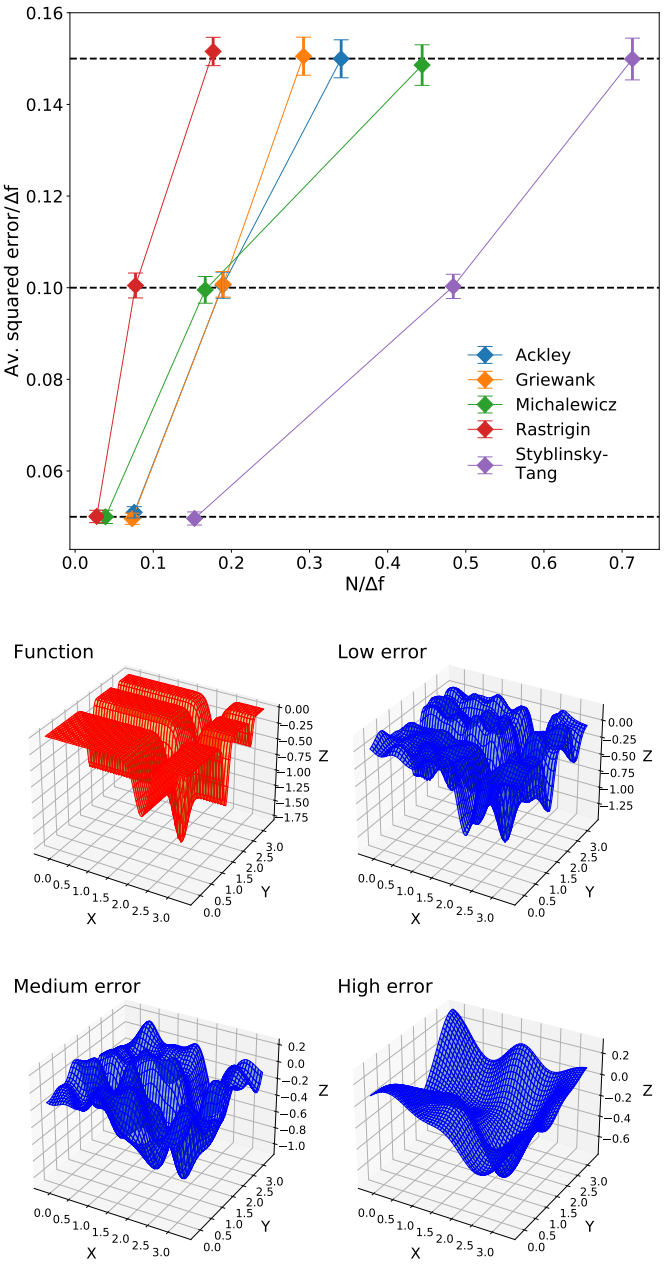


Fig. 1. Top: the error parameters used to generate predictors for each function. N has been divided by Δf to partially normalize across the benchmark. Bottom: example of predictors at the three error levels for the Michalewicz function.

We measure the quality of each method according to the *regret* of the optimal point x^{opt} found by the method, which is defined as $R(x^{\text{opt}}) = f(x^{\text{opt}}) - f_{\text{opt}}$. For the exclusion radius and discrepancy prediction methods we take x^{opt} to be the point that minimizes the mean of the surrogate model $\mathcal{M}$, while for the multi-fidelity method we take x^{opt} to be the point that minimizes the mean of the high-fidelity surrogate model.

Standard Bayesian optimization. The standard Bayesian optimization method is initialized with five random points, and is otherwise unaltered.

Exclusion radius. We initialize the exclusion radius method with zero real points and 50 predicted points. For this method we test a range of different values for the radius parameter r. Each benchmark has a different sized domain, so absolute values of r cannot be compared across different benchmark functions. For this reason, we select values of r relative to the size of the search space. Since each benchmark function has a square shaped search space (see Table 1), we use l to denote the side-length of this square, and we choose values of r so that $r/l = \{0.05,\ 0.1,\ 0.15,\ 0.2,\ 0.3\}$, meaning that we test settings of r that correspond to 5%, 10%, 15%, 20%, and 30% of the size of the search space.

In addition to this, we also test the case where $r/l = 0$, which corresponds to a zero radius, meaning that no points will be deleted during the optimization. This will allow us to compare the exclusion radius and the discrepancy prediction techniques against a baseline method that does not delete points.

Discrepancy prediction. The discrepancy prediction method is initialized with 45 predicted points and 5 randomly selected real points. Unlike the exclusion radius method, discrepancy prediction needs no other parameters. Both the exclusion radius and the discrepancy prediction methods were implemented on top of the package for Bayesian optimization GPyOpt [2].

The multi-fidelity method. The multi-fidelity method assumes that there is a hierarchy of lower-fidelity approximations of f that can be queried during the optimization. Each lower-fidelity approximation has an associated cost, with the idea that higher cost approximations give more accurate data, with f itself having the highest cost of all.

We benchmark against the most commonly used multi-fidelity approach [7, 10, 23, 25, 29] which assumes an autoregressive relationship between the lower fidelity models, and which uses one Gaussian process as surrogate model. This benchmark method uses a cost-sensitive version of the information gain acquisition function, as proposed by Swersky et al. and by Marco et al., which has been shown to be more efficient for multi fidelity optimization compared to expected improvement. We use an implementation provided by the python package Emukit [22].

Since we only have a single predictor, we apply the method to a hierarchy of two functions, with the real function f being the high-fidelity function, and the predictor p serving the low fidelity function. The acquisition function requires

that we assign costs to the two functions. So although our setup assumes that the predictor is essentially free to query, we were required to fix costs in order to apply the multi-fidelity method. We were unable to generate results for the "true" cost of f, which would correspond to setting the cost of f to be much higher than the cost of p, as this caused the method to almost exclusively query p, and the stopping condition of 80 real observations was not reached in a reasonable amount of time. Instead we set p to have cost 1, and we tested two values for the cost of f: 2 and 10.

The multi-fidelity method was initialized by using 5 random real points for the high-fidelity function, and 50 random points for the low-fidelity function.

5 Results

The results of our experiments are shown in Figure 2, which shows the results on a logarithmic scale, and they are also shown on a linear scale in Figure 5 in Appendix B. Both the exclusion radius and the discrepancy prediction methods outperform standard Bayesian optimization, especially during the early stages of the optimization process. This is particularly clear in the linear-scale charts. However in the logarithmic scale it can be seen that multi-fidelity methods eventually catch up once we are very close to an optimal point.

We also consider as a measure of performance the number of real observations needed to get within 5% of the optimal value. This percentage is quoted relative to the average value of the benchmark function: we define $\hat{f}$ to be the average value of f, computed by sampling 10000 points according to a grid design, and then we set our target as $0.05 \cdot \left(\hat{f} - f_{\text{opt}}\right)$. The plot in Figure 3 shows the first real observation at which each method achieves a regret that is better than this value. This data is also available in tabular form in Tables 4, 5, and 6 in the appendix.

Analysis. For the exclusion radius method, it can be seen that as the error level of the predictor increases, the optimal values for r/l also increase, indicating that a higher number of predicted points need to be discarded in higher error regimes.

Generally optimum values for the exclusion radius are between 0 and $0.15 \cdot l$ in the low error regime, between $0.05 \cdot l$ and $0.2 \cdot l$ in the medium error regime, and between $0.1 \cdot l$ and $0.2 \cdot l$ in the high error regime. In reality the level of error of the predictors is not known a priori, but the results shown in Figure 3 indicate that the choice $r/l = 0.1$ is suitable for all the three regimes.

Surprisingly, our benchmark test of setting the exclusion radius to $r/l = 0$, meaning that no points were deleted, was competitive, though not optimal, in the low and medium error regimes. However, the method performs very poorly in the high error regime. Hence, deleting points does have a positive effect on convergence speed.

The results also show that, for low and medium error predictors, the discrepancy prediction method is competitive with the exclusion radius technique on four out of the five benchmark functions, with only the Styblinski-Tang function

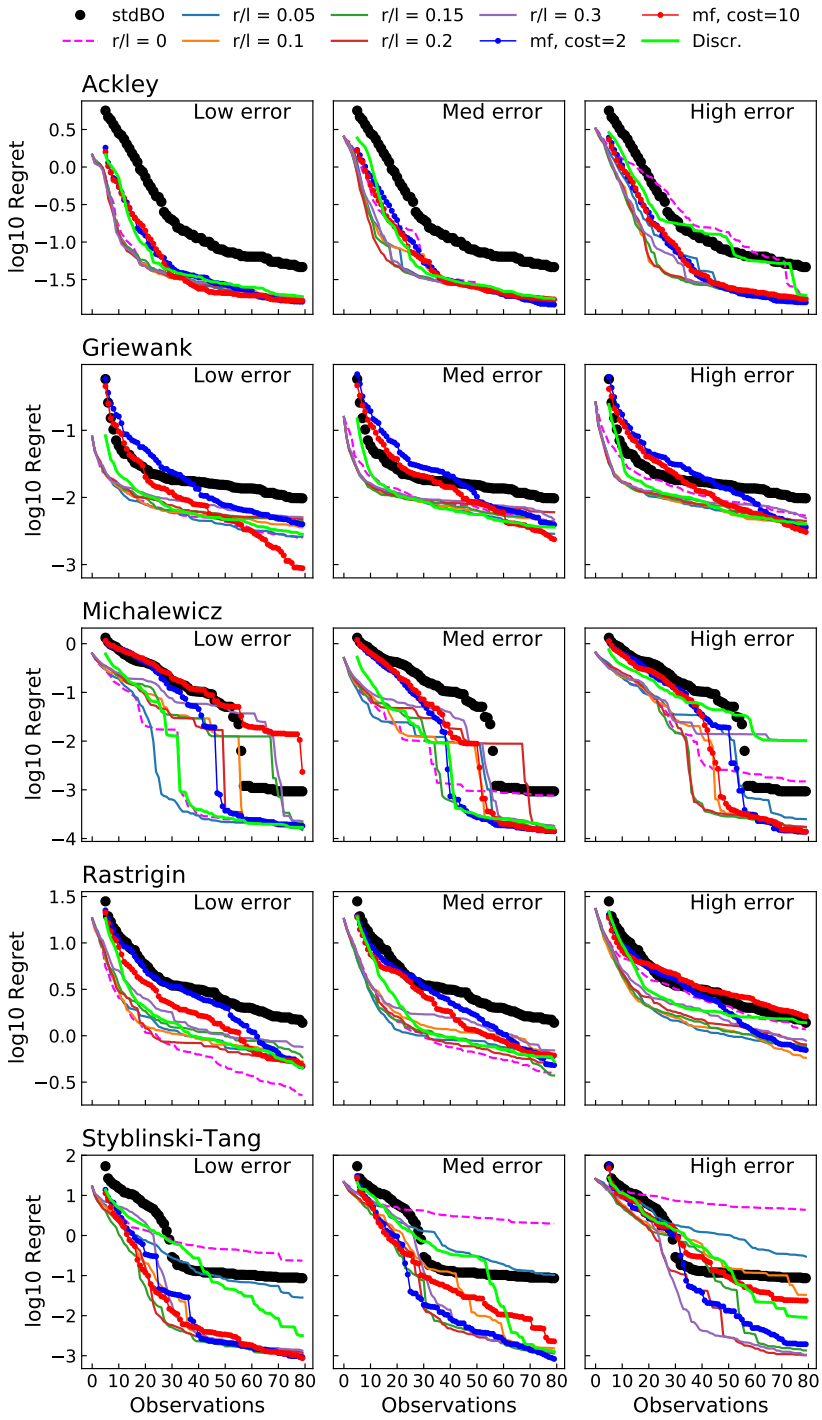


Fig. 2. Experimental results for all methods on all benchmarks. The curves for standard BO, discrepancy prediction, and the multi-fidelity Bayesian optimization experiments start at observation 5 as they are all initialized with 5 real points. The curves for the exclusion radius methods start from 0 as no real points are used for the initialization.

showing poorer convergence in the earlier stages of optimization. In the high error regime the method is less consistent.

To sum up, on our benchmark functions, if one has some idea about the error level of the predictor, then the parameter r can be fine-tuned to achieve excellent results. If the error level of the predictor is unknown, then either the exclusion radius method with $r/l = 0.1$, or the parameter-free discrepancy prediction method can be used, to achieve generally good results.

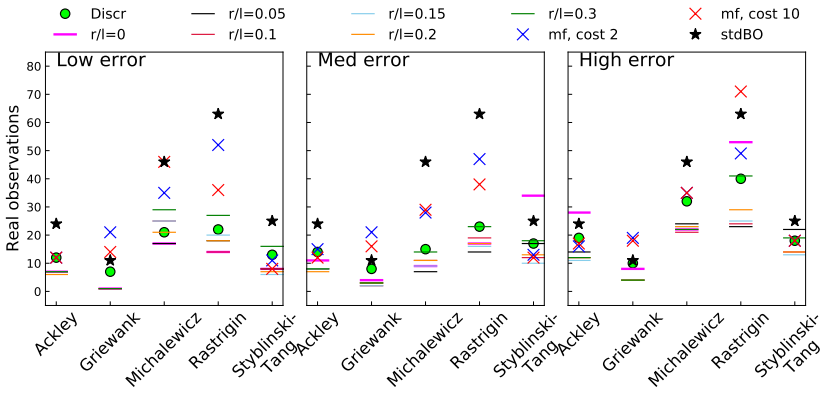


Fig. 3. The number of real observations needed to get within 5% of the optimal value.

Computation time. We also found that our methods were substantially faster in computation time when compared to the multi-fidelity approach. Figure 4 shows the wall clock time that was taken for each method to reach 80 real observations. The multi-fidelity approach can be seen to be substantially slower, and we found that this was for two reasons: the multi-fidelity approach uses more iterations, and each iteration takes substantially more time.

6 Conclusion

We have proposed two algorithms to accelerate Bayesian optimization by exploiting predicted knowledge. Both methods are conceptually simple and they are competitive with state of the art methods like multi-fidelity optimization, while requiring remarkably less computational time.

Experimentally, we found that a reasonable choice for the exclusion radius is $r/l = 0.1$, which is suitable for all the error levels that we considered. The discrepancy prediction method is overall less performant than the exclusion method, especially in the high error regime, but it has the advantage of not depending on any hyper-parameter.

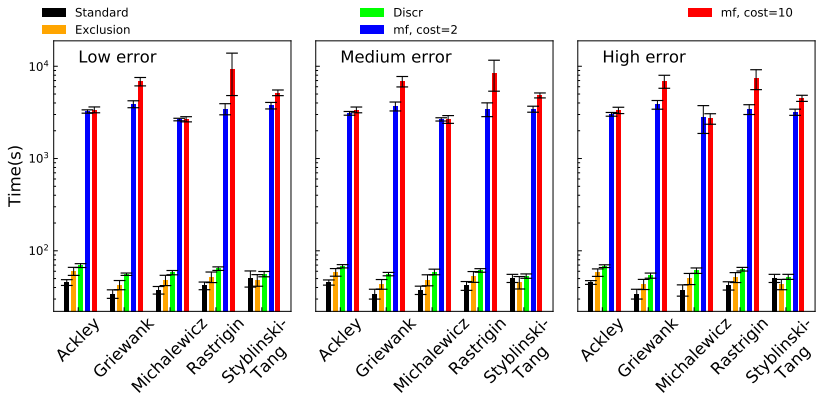


Fig. 4. Total computational time in seconds for each method.

References

1. Virtual library of simulation experiments:. https://www.sfu.ca/ ssurjano/rastr.html, accessed: 2021-02-25
2. authors, T.G.: Gpyopt: A bayesian optimization framework in python. http://github.com/SheffieldML/GPyOpt (2016)
3. Burger, B., Maffettone, P.M., Gusev, V.V., Aitchison, C.M., Bai, Y., Wang, X., Li, X., Alston, B.M., Li, B., Clowes, R., et al.: A mobile robotic chemist. Nature **583**(7815), 237–241 (2020)
4. Feurer, M., Springenberg, J.T., Hutter, F.: Using meta-learning to initialize bayesian optimization of hyperparameters. In: MetaSel@ ECAI. pp. 3–10. Citeseer (2014)
5. Forrester, A.I., Sóbester, A., Keane, A.J.: Multi-fidelity optimization via surrogate modelling. Proceedings of the royal society a: mathematical, physical and engineering sciences **463**(2088), 3251–3269 (2007)
6. Frazier, P.I.: A tutorial on bayesian optimization. arXiv preprint arXiv:1807.02811 (2018)
7. Ghoreishi, S.F., Allaire, D.: Multi-information source constrained bayesian optimization. Structural and Multidisciplinary Optimization **59**(3), 977–991 (2019)
8. Griffiths, R.R., Hernndez-Lobato, J.M.: Constrained bayesian optimization for automatic chemical design using variational autoencoders. Chem. Sci. **11**, 577–586 (2020)
9. Hennig, P., Schuler, C.J.: Entropy search for information-efficient global optimization. Journal of Machine Learning Research **13**(6) (2012)
10. Herbol, H.C., Poloczek, M., Clancy, P.: Cost-effective materials discovery: Bayesian optimization across multiple information sources. Materials Horizons (2020)
11. Huang, D., Allen, T.T., Notz, W.I., Miller, R.A.: Sequential kriging optimization using multiple-fidelity evaluations. Structural and Multidisciplinary Optimization **32**(5), 369–382 (2006)
12. Hse, F., Roch, L.M., Kreisbeck, C., Aspuru-Guzik, A.: Phoenics: A bayesian optimizer for chemistry. ACS Central Science **4**(9), 1134–1145 (2018)

13. Jamil, M., Yang, X.S.: A literature survey of benchmark functions for global optimisation problems. International Journal of Mathematical Modelling and Numerical Optimisation **4**(2), 150–194 (2013)
14. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. Journal of Global optimization **13**(4), 455–492 (1998)
15. Kandasamy, K., Dasarathy, G., Oliva, J.B., Schneider, J., Póczos, B.: Gaussian process bandit optimisation with multi-fidelity evaluations. In: Advances in Neural Information Processing Systems. pp. 992–1000 (2016)
16. Kennedy, M.C., O'Hagan, A.: Predicting the output from a complex computer code when fast approximations are available. Biometrika **87**(1), 1–13 (2000)
17. Kim, J., Kim, S., Choi, S.: Learning to warm-start bayesian hyperparameter optimization. arXiv preprint arXiv:1710.06219 (2017)
18. Lam, R., Allaire, D.L., Willcox, K.E.: Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In: 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. p. 0143 (2015)
19. Loh, W.L., et al.: On latin hypercube sampling. Annals of statistics **24**(5), 2058–2080 (1996)
20. Manteufel, R.: Evaluating the convergence of latin hypercube sampling. In: 41st Structures, Structural Dynamics, and Materials Conference and Exhibit. p. 1636
21. Močkus, J.: On bayesian methods for seeking the extremum. In: Optimization techniques IFIP technical conference. pp. 400–404. Springer (1975)
22. Paleyes, A., Pullin, M., Mahsereci, M., Lawrence, N., Gonzlez, J.: Emulation of physical processes with emukit. In: Second Workshop on Machine Learning and the Physical Sciences, NeurIPS (2019)
23. Patra, A., Batra, R., Chandrasekaran, A., Kim, C., Huan, T.D., Ramprasad, R.: A multi-fidelity information-fusion approach to machine learn and predict polymer bandgap. Computational Materials Science **172**, 109286 (2020)
24. Peherstorfer, B., Willcox, K., Gunzburger, M.: Survey of multifidelity methods in uncertainty propagation, inference, and optimization. Siam Review **60**(3), 550–591 (2018)
25. Poloczek, M., Wang, J., Frazier, P.: Multi-information source optimization. In: Advances in Neural Information Processing Systems. pp. 4288–4298 (2017)
26. Poloczek, M., Wang, J., Frazier, P.I.: Warm starting bayesian optimization. In: 2016 Winter Simulation Conference (WSC). pp. 770–781. IEEE (2016)
27. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N.: Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE **104**(1), 148–175 (2015)
28. Shields, B.J., Stevens, J., Li, J., Parasram, M., Damani, F., Alvarado, J.I.M., Janey, J.M., Adams, R.P., Doyle, A.G.: Bayesian reaction optimization as a tool for chemical synthesis. Nature **590**(7844), 89–96 (2021)
29. Song, J., Chen, Y., Yue, Y.: A general framework for multi-fidelity bayesian optimization with gaussian processes. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 3158–3167 (2019)
30. Styblinski, M., Tang, T.S.: Experiments in nonconvex optimization: stochastic approximation with function smoothing and simulated annealing. Neural Networks **3**(4), 467–483 (1990)
31. Swersky, K., Snoek, J., Adams, R.P.: Multi-task bayesian optimization. In: Advances in neural information processing systems. pp. 2004–2012 (2013)
32. Williams, C.K., Rasmussen, C.E.: Gaussian processes for machine learning, vol. 2. MIT press Cambridge, MA (2006)

33. Wu, J., Toscano-Palmerin, S., Frazier, P.I., Wilson, A.G.: Practical multi-fidelity bayesian optimization for hyperparameter tuning. In: Uncertainty in Artificial Intelligence. pp. 788–798. PMLR (2020)

A Experimental Parameters

Method	**# Real**	**# Pred.**	$\hat{A}/\Delta f$	r/l	**Cost**
Standard	5	0	—	—	—
Exclusion radius	0	50	0.05, 0.1, 0.15, 0.2	0, 0.05, 0.1, 0.15, 0.2, 0.3	—
Discrepancy prediction	5	45	0.05, 0.1, 0.15, 0.2	—	—
Multi-fidelity	5	50	0.05, 0.1, 0.15, 0.2	—	2, 10

Table 2. Experimental setup for each method. # Real and # Pred. denote the number of real and predicted points used to initialize the method. $\hat{A}/\Delta f$ denotes the predictor accuracies that were tested. r/l gives the values of the radius parameter for the exclusion radius technique, while cost denotes the costs that were tested for the multi-fidelity technique.

Function	Error parameters (N)			Exclusion radius (r)					
	5%	10%	15%	0%	5%	10%	15%	20%	30%
Ackley	0.83	2.8	3.75	0	0.4	0.8	1.2	1.6	2.4
Griewank	0.12	0.31	0.48	0	1.0	2.0	3.0	4.0	6.0
Michalewicz	0.07	0.3	0.8	0	0.16	0.31	0.47	0.63	0.94
Rastrigin	2.23	6.22	14.22	0	0.51	1.02	1.54	2.05	3.07
Styblinski-Tang	50.18	158.91	234.18	0	0.5	1.0	1.5	2.0	3.0

Table 3. Absolute values of the set up parameters for the exclusion radius method and for each benchmark functions.

B Regrets Versus Real Observations on a Linear Scale

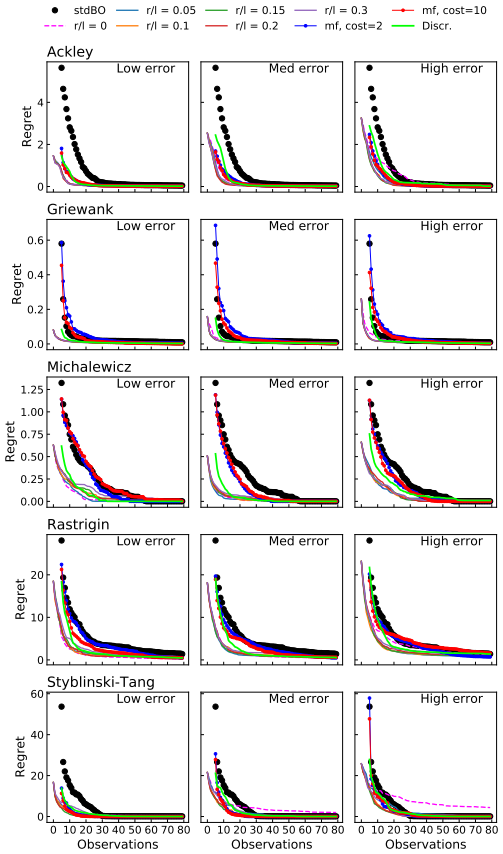


Fig. 5. This shows the curves from Figure 2 on a linear scale.

C Convergence to a 5% target

Low Error						
Function	r/l	Exc. radius	Standard	Discr.	MF cost 2	MF cost 10
Ackley	0	7	24	12	12	12
	0.05	7				
	0.1	7				
	0.15	7				
	0.2	**6**				
	0.3	7				
Griewank	0	**1**	11	7	21	14
	0.05	**1**				
	0.1	**1**				
	0.15	**1**				
	0.2	**1**				
	0.3	**1**				
Michalewicz	0	**17**	46	21	35	46
	0.05	**17**				
	0.1	25				
	0.15	25				
	0.2	21				
	0.3	29				
Rastrigin	0	**14**	63	22	52	36
	0.05	18				
	0.1	**14**				
	0.15	20				
	0.2	18				
	0.3	27				
Styblinski-Tang	0	8	25	13	11	8
	0.05	8				
	0.1	8				
	0.15	**6**				
	0.2	7				
	0.3	16				

Table 4. Number of steps needed to get within 5% of the optimal point for low error predictors.

Medium Error						
Function	r/l	Exc. radius	Standard	Discr.	MF cost 2	MF cost 10
Ackley	0	11	24	14	15	12
	0.05	8				
	0.1	8				
	0.15	8				
	0.2	**7**				
	0.3	8				
Griewank	0	4	11	8	21	16
	0.05	3				
	0.1	**2**				
	0.15	**2**				
	0.2	3				
	0.3	3				
Michalewicz	0	9	46	15	28	29
	0.05	**7**				
	0.1	11				
	0.15	9				
	0.2	11				
	0.3	14				
Rastrigin	0	17	63	23	47	38
	0.05	**14**				
	0.1	19				
	0.15	16				
	0.2	17				
	0.3	23				
Styblinski-Tang	0	34	25	17	13	12
	0.05	17				
	0.1	12				
	0.15	**10**				
	0.2	13				
	0.3	18				

Table 5. Number of steps needed to get within 5% of the optimal point for medium error predictors.

High Error						
Function	r/l	Exc. radius	Standard	Discr.	MF cost 2	MF cost 10
Ackley	0	28	24	19	16	17
	0.05	14				
	0.1	12				
	0.15	**11**				
	0.2	12				
	0.3	12				
Griewank	0	8	11	10	19	18
	0.05	**4**				
	0.1	**4**				
	0.15	**4**				
	0.2	**4**				
	0.3	**4**				
Michalewicz	0	22	46	32	35	35
	0.05	24				
	0.1	**21**				
	0.15	23				
	0.2	23				
	0.3	22				
Rastrigin	0	53	63	40	49	71
	0.05	**23**				
	0.1	24				
	0.15	25				
	0.2	29				
	0.3	41				
Styblinski-Tang	0	-	25	18	18	18
	0.05	22				
	0.1	14				
	0.15	**13**				
	0.2	14				
	0.3	19				

Table 6. Number of steps needed to get within 5% of the optimal point for high error predictors.

Argumentation in Trust Services within a Blockchain Environment

Liuwen Yu[1,3,4][0000−0002−7200−6001], Mirko Zichichi[2,3,4][0000−0002−4159−4269], Réka Markovich[1][0000−0002−2488−2293], and Amro Najjar[1][0000−0001−7784−6176]

[1] University of Luxembourg, Luxembourg
[2] Universidad Politécnica de Madrid, Spain
[3] University of Bologna, Italy
[4] University of Turin, Italy

Abstract. Both argumentation and trust concern multi-lateral uncertainties, while argumentation owns the ability to enhance trust in many ways. In the field of trust service where the trustee administers financial assets on behalf of principals, trust is an indispensable element. Often, the trustees withhold the investment plans and of which the decision-making process from their principals such that these services lack of transparency documentation, traceability, and inclusive decision-making mechanisms. In this paper, we integrate formal argumentation within a blockchain framework. Both argumentation and blockchain have distinctive features that complement each other. They together make the decision-making of the trustees transparent and traceable in order to gain trust and confidence in principals. We introduce three possible architectures and we evaluate and compare them considering different technical, financial, and legal aspects. Specifically, we discuss the role of argumentation in building trust between trustees and their principals.

Keywords: trust services · argumentation · negotiation · blockchain · smart contracts · artificial intelligence

1 Introduction

Trust service is concerned as persons or organization that acts on behalf of another person or persons to deal with the tasks involves finances, i.e., managing the assets, where trust from trustors plays a crucial role in entering into the contractual relations with trustee. Fund management, as a strand of trust services, is meant that the fund managers, i.e., trustees, are in the position of a fiduciary and put their principals' interest ahead of their own to construct a portfolio of securities (e.g., stock, bonds, mutual funds, etc.), with a duty to preserve good faith and trust. In general, fund management mainly has a two-stage procedure. At the first stage fund managers are supposed to perform an evaluation of the selected securities on account of their expertise. At a second stage, the transactions based on the first stage are executed. As a matter of course, trust problem will emerge in both stages. On the one hand, the seeds of distrust of such fiduciary may be planted from the difference between the principal's and the fund managers' expertise, as well as the reservation and lack of documentation of the decision-making process of

investment plans. The legislators have already taken this problem into account, they can (and does[5]) declare the principal's right to check the fiduciary's relevant activities in order to give weight to this duty by its intended controllability. On the other hand, weather the transactions are executed as planed is also the original of distrust.

In this study, we propose an integrated framework that incorporates formal argumentation within a blockchain environment for making the decision-making processes of fund management transparent and traceable. As suggested by both academics and industries, smart contracts within blockchain technology can also be engaged in the core activities in the securities market [24,61], proven by the surge of Decentralized Finance (DeFi) [68]. The involvement of smart contracts and blockchain can address the second concern, i.e., make the transactions transparent and auditable. Nevertheless, blockchain for transactions alone does not address the first trust problem, it is actually used only to trace the output of such a decision-making process. The principals still don't have access to why the given transaction happened and whether it happened indeed in his best interest. To this sense, trust can be understood as a relational attribute between a social actor and /or institutions [8], and trust is also a technique for dealing with uncertainty regarding other parties' actions and communications [31]. We argue that formal argumentation and trust share a common function: they both deal with changes and uncertainties in complex social environment [45]. We aim to show that formal argumentation is suited for modelling the decision-making process of fund management, which is multi-lateral interaction and reasoning based on incomplete and inconsistent information to help explain why a claim or a decision is made. In the fund management case, information incorporating the different fund managers' opinions is provided by different conflict-resolution techniques: argumentation is used to decide whether to buy, sell or hold securities, and negotiation to determine the quantities and investment timing, and thus to provide explanations. By integrating argumentation with blockchain, a reasoning system put in place for making these decisions could be featured with auditability, transparency, traceability and explainability, which all serve to enhance reliability and trust—in such an industry which is named after it.

Our proposal is a framework integrating different methodologies based on different considerations:

(i) First we consider the ecosystem of the trust services, the fund management (at the securities market) to see the roles of the parties and their relation, especially that of the fund managers. This is what we start with in Section 2 as a motivation.
(ii) The technical environment for the solution we propose is blockchain and smart-contracts given that their application in the trading itself on the securities market is rising [24,51]. The expertise of the fund managers and their decision based on that triggers the transaction, that is, the smart contract's execution. The interface giving external input needed for the smart contract's execution—for some reason, for instance, the human expertise's irreplaceability—is called (blockchain) oracle. The blockchain systems and their reliance on oracles involve some considerations

[5] For instance, the 6:315. § of the Hungarian Civil Code (Act V of 2013) says: *The principal and the beneficiary shall have the right to check the fiduciary's activities relating to asset management.*

the understanding of which is needed for the proper involvement of the methodologies we propose, thus we introduce shortly what oracles are in the blockchain environment and how they are supposed to work in Section 3.

(iii) In order to optimize the involved expertise of investing the principal's money, we count with more than one fund managers. These fund managers might, of course, have different opinions about selling or buying, what, when and how much. However, at the end of the day, they need one decision: the smart contract needs one input. To optimize this decision-making process, its traceability in the computational environment and its integration into the blockchain environment, we propose using formal argumentation and negotiation in the multi-agent systems setup. To have this paper self-contained, we introduce these methodologies and discuss their relevance and applicability in the current process in Section 4.

(iv) Integrating formal argumentation and multi-agent negotiation for creating the proper external input triggering the transaction's smart contract leads us to the framework we call Intelligent Human-input-based Blockchain Oracle (IHiBO). We consider three possible architectures in the blockchain framework, for each we have a different way to integrate argumentation and negotiation in the set of blockchain framework, evaluate and compare them regarding different technical and legal aspects in Section 5, we don't only consider traceability, verifiability, execution overhead costs and possible failure, but also the trade secret, and privacy issues related to each architecture.

Afterwards, we give the discussion, and not only the related works but also the consideration of our contribution and future perspectives.

2 Motivation

In this section, we generally talk about the procedure of fund management (at the securities market) and the roles of the parties and their relation, in order to show that the decision-making process can be suited into argumentation modeling.

Fund managers play an important role in the investment and financial world, they provide investors with peace of mind, knowing their money is in the hands of an expert [11]. However, the reality is not always as one wished, investors tend to know but they don't, in reality, where their money goes, why, and how much is the real profit. In portfolio management, the core duties of fund managers under AIFMD[6] and UCITSD[7] is to perform portfolio management and risk management on behalf of their investors. The fund can be managed by one person, by two people as co-managers, or by a team of three or more people. Fund managers primarily research and determine the best stocks, bonds, or other securities to fit the strategy of the fund, then buy and sell them. Since the fund managers are responsible for the success of the fund, they must also research

[6] Directive 2011/61/EU of the European Parliament and of the Council of 8 June 2011 on Alternative Investment Fund Managers (AIFMD.http://data.europa.eu/eli/dir/2011/61/oj)

[7] Directive 2009/65/EC of the European Parliament and of the Council of 13 July 2009 on the coordination of laws, regulations and administrative provisions relating to undertakings for collective investment in transferable securities (UCITS). http://data.europa.eu/eli/dir/2009/65/oj

companies, and study the financial industry and the economy. Keeping up to date on trends in the industry helps the fund managers make key decisions that are consistent with the fund's goals [15]. The main characteristic of investing in a fund is trusting the investment management decisions to the professionals.

The process of portfolio management on the manager side is formally defined as follows [16][17]: *Portfolio management is a dynamic decision process, whereby a business's list of active new product (and development) projects is constantly up-dated and revised. In this process, new projects are evaluated, selected and prioritized; existing projects may be killed or de-prioritized. The portfolio decision process is characterized by uncertain and changing information, dynamic opportunities, multiple goals and strategic considerations, multiple decision-makers and locations. The portfolio decision process encompasses or overlaps a number of decision-making processes within the business, making Go/Kill decisions on individual projects on an on-going basis, and developing a new product strategy for the business.*

A possible simplified process of fund investment management includes the following activities. Firstly, the investors pool their money together. Then fund managers gather information and conduct investment research, prepare the specific plan for the investment portfolio. According to their research and the final decision of investment plan, fund managers invest securities on behalf of their clients (investors). The investment generates returns and the returns would be passed down to investors.

3 Formal Argumentation and Negotiation

In section 2, we show that fund managers conduct the securities transactions directly, such behavior creates a sense of insecurity in clients, how and why the fund managers make the investment plans and actions need to be explained and modeled. In the second move of fund management described in Fig.1, various managers might have different investment plans based on their own expertise and research that may conflict with each other. We present the solution proposals in this section to resolve the conflicts by formal argumentation and negotiation.

Formal argumentation or computational argumentation in artificial intelligence (AI) is a formalism for representing and reasoning with incomplete and inconsistent information. A wide variety of reasoning and dialogical activities can be captured by argumentation models in a formal and still quite intuitive way, allowing the integrationof different concrete techniques and the development of applications that humans can trust. Dung's work in 1995 illustrates an argumentation system consisting of a set of arguments and the relation (attacks) between them [21]. Argumentation semantics are defined later by Baroni and Giacomin for gathering acceptable arguments lying on different criterias [7], in a way that somehow emulates the way humans tackle such a complex task [1,5,50]. Formal argumentation also can be used for modeling the dynamic interactions among agents which is particularly at stake in a multi-agent context: the system evolves as the agents put forward new arguments or retract arguments and relations [10,19,36]. There are lots of variants of Dung's original framework, extending the theory with preference [2,30], support [14,71,72], probabilities [27,35], etc. In this section, we use agent ab-

stract argumentation which is introduced in one of the authors' latest work [70], and autonomous negotiation for dealing with conflicting information raised by agents.

3.1 Agent Argumentation

We generalize argumentation frameworks studied by Dung (1995), which are directed graphs, where the nodes are arguments, and the arrows correspond to the attack relation.

Definition 1 (Argumentation framework [20]). *An* argumentation framework *(AF) is a pair $\langle \mathscr{A}, \rightarrow \rangle$ where $\mathscr{A}$ is a set called arguments, and $\rightarrow \subseteq \mathscr{A} \times \mathscr{A}$ is a binary relation over $\mathscr{A}$ called attack. For a set $S \subseteq \mathscr{A}$ and an argument $a \in \mathscr{A}$, we say that S attacks a if there exists $b \in S$ such that b attacks a, a attacks S if there exists $b \in S$ such that a attacks b, $a^- = \{b \in \mathscr{A} | b \text{ attacks } a\}$, $S^-_{out} = \{a \in \mathscr{A} \backslash S | \text{ a attacks } S\}$.*

Dung's admissibility-based semantics is based on the concept of defense. A set of arguments defends another argument if they attack all its attackers.

Definition 2 (Admissible [20]). *Let $\langle \mathscr{A}, \rightarrow \rangle$ be an AF. $E \subseteq \mathscr{A}$ is* conflict-free *iff there are no arguments a and b in E such that a attacks b. $E \subseteq \mathscr{A}$* defends *c iff for all arguments b attacking c, there is an argument a in E such that a attacks b. $E \subseteq \mathscr{A}$ is* admissible *iff it is conflict-free and defends all its elements.*

For their principle-based analysis, Baroni and Giacomin define semantics as a function from argumentation frameworks to sets of subsets of arguments.

Definition 3 (Dung semantics [7]). *A Dung semantics is a function σ that associates with an argumentation framework $AF = \langle \mathscr{A}, \rightarrow \rangle$, a set of subsets of $\mathscr{A}$, the elements of $\sigma(AF)$ are called extensions.*

Dung distinguishes several definitions of extension.

Definition 4 (Extensions [20]). *Let $\langle \mathscr{A}, \rightarrow \rangle$ be an AF. $E \subseteq \mathscr{A}$ is a* complete extension *iff it is admissible and it contains all arguments it defends, i.e., $E = \{a | E \text{ defends } a\}$. $E \subseteq \mathscr{A}$ is a* grounded extension *iff it is the smallest (for set inclusion) complete extension. $E \subseteq \mathscr{A}$ is a* preferred extension *iff it is a largest (for set inclusion) complete extension. $E \subseteq \mathscr{A}$ is a* stable extension *iff it is conflict-free and it attacks each argument which does not belong to E.*

Each kind of extension may be seen as an acceptability semantics that formally rules the argument evaluation process. In this article, we use $\sigma \in \{c, g, p, s\}$ to represent Dung semantics {complete, grounded, preferred, stable}.

An agent argumentation framework extends an argumentation framework with a set of agents and a relation associating arguments with agents. Note that an argument can belong to one agent or multiple agents.

Definition 5 (Agent argumentation framework [72]). *An* agent argumentation framework *(AAF) is a 4-tuple $\langle \mathscr{A}, \rightarrow, \mathscr{S}, \sqsubset \rangle$ where $\mathscr{A}$ is a set of arguments, $\rightarrow \subseteq \mathscr{A} \times \mathscr{A}$ is a binary relation over $\mathscr{A}$ called attack, $\mathscr{S}$ is a set of agents or sources, $\sqsubset \subseteq \mathscr{A} \times \mathscr{S}$ is a binary relation associating arguments with agents. $\mathscr{A}_\alpha = \{a \in \mathscr{A} | a \sqsubset \alpha\}$ for all arguments that belong to agent α, $\mathscr{S}_a = \{\alpha | a \sqsubset \alpha\}$ for all agents that have argument a.*

Social agent semantics [70] For the decision making of fund management, we use so-called social semantics, which is based on a reduction to preference-based argumentation by for each argument counting the number of agents that have the argument. It thus interprets agent argumentation as a kind of voting, as studied in social choice theory or judgment aggregation, this is also closed to fund management.

We first give the definition of a preference-based argumentation framework.

Definition 6 (Preference-based argumentation framework [30]). *A preference-based argumentation framework (PAF) is a 3-tuple $\langle \mathscr{A}, \rightarrow, \succ \rangle$ where $\mathscr{A}$ is a set of arguments, $\rightarrow \subseteq \mathscr{A} \times \mathscr{A}$ is a binary attack relation, $\succ$ is a partial order (irreflexive and transitive) over $\mathscr{A}$, called preference relation.*

There are two different reductions of preference being first introduced[3], after which there are two more reductions [63]. We refer to those papers for an explanation and motivation, while users should select one reduction according to their particular application, one can refer to the principle-based approach to distinguish these reductions [63,70].

Definition 7 (Reductions of PAF to AF (PR)). *Given an $PAF = \langle \mathscr{A}, \rightarrow, \succ \rangle$:*

- $PR_1(PAF) = \langle \mathscr{A}, \rightarrow' \rangle$, *where* $\rightarrow' = \{a \rightarrow' b | a \rightarrow b, b \not\succ a\}$.
- $PR_2(PAF) = \langle \mathscr{A}, \rightarrow' \rangle$, *where* $\rightarrow' = \{(a \rightarrow' b | a \rightarrow b, b \not\succ a \text{ or } b \rightarrow a, \text{ not } a \rightarrow b, a \succ b\}$.
- $PR_3(PAF) = \langle \mathscr{A}, \rightarrow' \rangle$, *where* $\rightarrow' = \{(a \rightarrow' b | (a \rightarrow b, b \not\succ a \text{ or } a \rightarrow b, \text{ not } b \rightarrow a\}$.
- $PR_4(PAF) = \langle \mathscr{A}, \rightarrow' \rangle$, *where* $\rightarrow' = \{a \rightarrow' b | a \rightarrow b, b \not\succ a, \text{ or } b \rightarrow a, \text{ not } a \rightarrow b, a \succ b, \text{ or } a \rightarrow b, \text{ not } b \rightarrow a\}$.

In social agent semantics, an argument is preferred to another argument if it belongs to more agents. The reduction from AAF to PAF is used as an intermediary step for social agent semantics.

Definition 8 (Social Reductions of AAF to PAF (SAP)). *Given an $AAF = \langle \mathscr{A}, \rightarrow, \mathscr{S}, \sqsubset \rangle$, $SAP(AAF) = \langle \mathscr{A}, \rightarrow, \succ \rangle$ with $\succ = \{a \succ b || \mathscr{S}_a| > |\mathscr{S}_b|\}$.*

Definition 9 (Social Reductions of AAF to AF (SR)). *Given an $AAF = \langle \mathscr{A}, \rightarrow, \mathscr{S}, \sqsubset \rangle$, $SR_i(AAF) = PR_i(SAP(AAF))$, PR_i is one of the four reductions of PAF to AF, where the semantics $\delta(AAF) = \sigma(SR_i(AAF)) = \sigma(PR_i(SAP(AAF)))$ for $i \in \{1,2,3,4\}$.*

3.2 Autonomous Agents and Negotiation

A software agent is a software that acts on behalf of another actor (often a human user) to perform a task or achieve a given goal [69]. Agents are designed to be bound to individual perspectives [58]. This makes agents good candidates to represent the subjectivity and nuances of different expert opinions. Multi-agent systems [66] provide a distributed platform capable of implementing intelligence in decentralized ecosystems such as blockchain-based systems where agents are capable, using well-established conflict-resolution mechanisms (e.g. negotiation), of helping the different stakeholders finding agreements that satisfy their often conflicting interests.

In his influential book, Dean Pruitt provides one of negotiation's most widely accepted definitions: "Negotiation is the process by which a joint decision is made by two or more parties. The parties first verbalize contradictory demands and then move towards agreement by a process of concession making or search for new alternatives" [52]. The problem being negotiated, or the topic under discussion (e.g. car purchase) can be usually divided into issues (also called attributes). Some negotiations involve only single issue (e.g. car price) whereas others involve multiple issues (e.g. price and delivery time). Negotiators may not only disagree on the value assigned to each issue, the priority given to each issue can differ from one negotiator to another and hence this can be a source of both divergence and convergence [54]. Automated negotiation is one taking place among autonomous agents [28]. Autonomous negotiation has a protocol. The latter is the set of rules that governs the interactions during a negotiation session (also called a thread). Whereas the negotiation protocol defines what is the set of possible actions that can be taken during a negotiation session, an agent has a decision model [23,40] that allows the agent to (i) evaluate the value of an offer received from the opponent (e.g., using a utility function), (ii) decide whether it is acceptable (also called acceptance condition [6]), and (iii) determine what to do next (known as the negotiation strategy [23]). Automated negotiation has been applied to solve conflicts and reach agreements in several domains including cloud and service provisioning [41], smart grid and power distribution [62], and trading and stock market [67].

3.3 Conflict Resolution

The process of portfolio management fits well with argumentation theory in artificial intelligence. The decision can be seen as being based on arguments and counter-arguments. Argumentation, as the result, can be useful for deriving decisions and explaining a choice already made. Managers provide their arguments from their own research to identify promising stocks with different level of accuracy and thereby make different portfolio choices which are likely to be incomplete and inconsistent.

The fictitious simple example (the real life cases would be much more complex) is as follows. Manager α and β hold the arguments *a*: *To buy the stocks, since the company just donated to charities that is beneficial to good commercial reputation*, while another manager γ at the same time is against to buy the stocks, he holds the arguments b_1 and b_2, b_1 is *To sell the stocks, since there is evidence that the leader is under accusations of charity fraud*, and b_2 is *To sell the stocks, since the company has poor sales performance.* However, manager α brings out the argument c_1 *The official has clarified the accusations collapsed*, and β brings c_2 *The company is going to adopt a new technology which will bring huge benefit.*

Based on the above, we can build an agent argumentation framework on the left side of Fig.1, $AAF = \langle \mathscr{A}, \rightarrow, \mathscr{S}, \sqsubset \rangle$ where $\mathscr{A} = \{a, b_1, b_2, c_1, c_2\}$, $\rightarrow = \{(b_1, a),\ (b_2, a),\ (c_1, b_1),\ (c_2, b_2), (a, b_1), (a, b_2)\}$, $\mathscr{S} = \{\alpha, \beta, \gamma\}$, $\sqsubset = \{(a, \alpha),\ (a, \beta),\ (b_1, \gamma),\ (b_2, \gamma),\ (c_1, \alpha),\ (c_2, \beta)\}$. Since $|\mathscr{S}_a|$ it the most preferrred, we get the corresponding PAF where $a \succ b_1, b_2, c_1, c_2$, and giving the four reductions from PAF to AF, we have the only AF on the right side (without the preference below) of Fig.1. Then we can calculate the only acceptable set $\{a, c_1, c_2\}$. The set tells the final decision is to buy the stocks. One thing needs to be noticed: argumentation does not always provide a definite

outcome. Depending on the decision making process, different protocols can be specified in advance for such cases: e.g. to roll back or to assign weights to the arguments and the relation among them (so that these cannot be always equal).

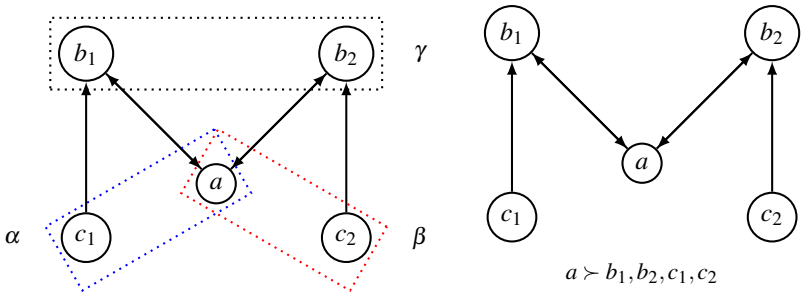


Fig. 1. Social reduction

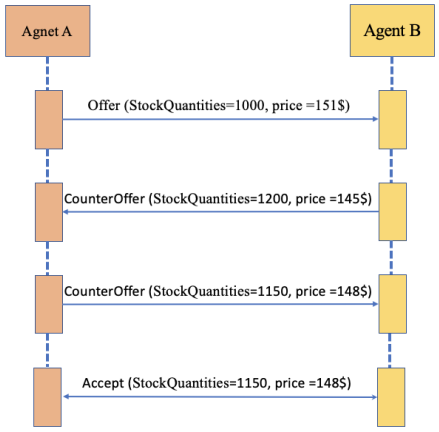


Fig. 2. Negotiation Sequence to Decide The Quantities and The Price

After deciding to sell the stocks, the next problem is the numbers of stocks to sell and the sell timing. Here the computational automated negotiation comes into play. To illustrate how it works, we give an example of the negotiation sequence based on the on the quantities of stocks to sell. The negotiation process is based on the alternating offer protocol [55]. Agents can bid new offers to the opponent ($Offer()$ function). When receiving an offer, and agent can accept it using $accept()$ function or reject it and propose a counter-offer (with the $CounterOffer()$ function). In the example, we have

a manager *A*, i.e., agent *A*, and manager *B*, i.e., agent *B*. Agent *A* proposes to sell 1000 stocks at the price of 151$, while agent *B* counteroffers to sell 1200 stocks at the price of 145$, then agent *A* proposes to sell 1150 stocks at the price of 148$. The final offer given by *A* is accepted by both parties which means they come to an agreement.

4 Blockchain in Financial Agreements and Architectures

The tamper-resistant property of DLTs enables a favourable environment for storing information that can be later audited. For the fund management use case we are dealing with, we refer to a generic smart contract based operation of security transaction, implemented using different kinds of the systems/technologies. In this subsection, we outline the potential of distributed ledger technologies (DLTs) to revolutionize financial agreements and a particular instance of how fund managers trade securities on behalf of their clients on blockchain platform.

4.1 Distributed Ledger Technologies

There is a growing body of work generated on the design and utilization principles for blockchain and DLTs [33]. The underlying premise of blockchain and its various applications is the elimination of untrustworthy third parties such that the users themselves are the authority of maintaining the ledgers which are immutable. The immutability of blockchains also enhances the distributed trust since it is nearly impossible to tamper any transactions stored in blockchains and all the historical transactions are auditable and traceable [73]. In the case of the blockchain, the ledger is organized into chronologically ordered blocks where each block is sequentially linked to the previous one [42]. When the majority of network nodes execute the exact same protocol, such as in the Bitcoin network, the blockchain is cryptographically guaranteed to be tamper-proof and unforgeable. A feature that some DLTs enable is the possibility to execute smart contracts, firstly introduced by the Ethereum blockchain [13], which is reshaping the conventional commercial industries [32,73,74]. Smart contracts consist of instructions that, once deployed on the ledger, cannot be altered and thus allowing the outcome of their execution to be always the same for anyone who runs it (i.e. the DLT network nodes). Usually, the possible instructions of a smart contract are embedded in the DLT protocol and their execution can only involve data coming from other smart contracts or from the user's inputs, e.g. smart contracts cannot fetch a webpage on the Internet. This "closure" ensures the execution of smart contracts to be more resistant to attacks with a higher degree of certainty, thus making the whole system more secure [73]. However, it also leads to a very restricted use case where DLTs are actually closed networks like a computer with no Internet connection. This obviously limits the possible usage of these technologies, since the vast majority of the possible smart contract applications would require real-time information from the network external world.

In order for smart contracts to operate in the real world, data must flow in both directions and thus the high demand for applications gave birth to blockcahin oracles. These third-party systems act as a bridge that connects the DLT network and the "outside" world, providing the ability to retrieve, verify and digest the data into smart contracts.

Oracles can be implemented as: (i) software, by far the most widely used, they interact with the information needed from online sources; (ii) hardware, retrieve data from the physical world directly through scanners sensors; (iii) human, interacting with individuals. In all cases their off-chain execution is either centralized, i.e. coming from a single source, or decentralized, consensus-based multitude of sources [9].

4.2 Decentralized Finance

Both scholars and industries have examined the commercial implications of DLTs and smart contracts, for instance, the financial services of tokenized securities settlement and clearing[29]. The advent of DLTs has the potential to restructure this paradigm by breaking the stigma, only apparently immutable, of centrality and of central counterparties (CCPs) [51]. Decentralized Financial Market Infrastructures (dFMI) [24] *are consortium entities whose members are comprised of the main participants in a market, organized in a peer-to-peer model, which is governed by dFMI participants themselves rather than a central intermediary.* In some applications smart contracts can take on a role similar to that previously played CCPs, e.g. acting as a margin calculating agent and taking on the task of transferring collateral. Although in a different way, the smart contract can be used to resolve disputes in the event of non-compliance with payment [39]. Alternatively, smart contracts can support the central counterparty, which can maintain the business model by leveraging the blockchain to calculate and update collateral as well as manage funds, thus relying on financial cryptography. A concrete application of DLTs for the trading of securities by fund managers is Lianjiaorong, a blockchain AssetBacked securitization platform, built by the Bank of Communications in China [47]. The blockchain is maintained by original stock holders, trust companies, investors, rating agencies, accountants, lawyers, regulators and it links funds and assets on the ledger, realizing the credit penetration of the securities business system.

4.3 Blockchain Architectures

In this subsection we deal with the operation that is the outcome of the negotiation and argumentation processes seen above, i.e. the decision, that is given as input to a smart contract, e.g., buy a stock. We refer to this smart contract as the "TransactionSC". In the following, we compare three different architectures that can take form in our blockchain framework, for reaching the decision to give as input to the TransactionSC (Figure 3). We take as reference Table 1 for comparing the three architectures.

1. Centralized Oracle The first architecture we consider is the simplest one, where argumentation and negotiation phases do not involve any blockchain process, neither a smart contract execution. These are executed in a "centralized" environment, e.g. a web platform or an internal firm application. Each decision coming after the negotiation will be given as input to the TransactionSC by a single service in this environment, that provides the role of an oracle.

A discriminating factor in choosing one architecture over another is where the information needed for execution is stored. In the case of this architecture, i.e. using a

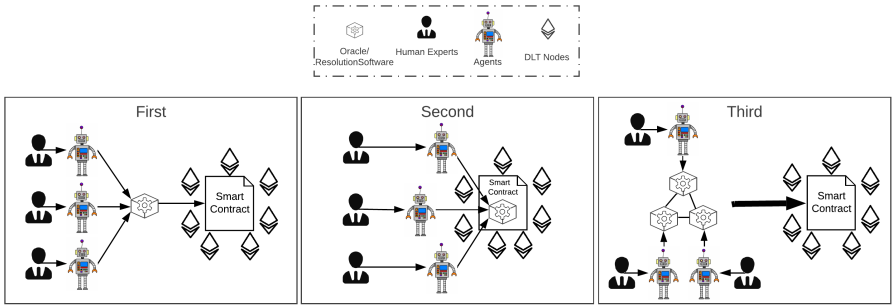


Fig. 3. IHiBO with three architectures

classical centralised oracle, the complete execution of the conflict resolution would be scarcely verifiable, because only the results would be stored on the blockchain. It would also be highly susceptible to a single point of failure.

2. Smart Contract Argumentation and Negotiation In the second architecture, argumentation and negotiation are directly implemented as smart contracts, and thus are executed following the blockchain protocol. It means that the human experts, through their agent software, directly interact with the blockchain for giving in input the data for constructing the argumentation graph and then for enacting the negotiation functions that are expressed as smart contract instructions.

The argumentation graph (and all the data needed for execution too) is necessary for the execution of the whole process, so it is constantly updated. This information only needs to be stored on the ledger in the case of this architecture. The disadvantages of storing large amounts of data on-chain are many, mainly, the high transaction cost [32] and the almost impractical deployment latency [75]. However, the advantage of this architecture is that trade execution would be fully tracked and verifiable, as execution would be done completely through smart contracts in the blockchain.

3. Decentralized Oracle Finally, the third architecture we consider consists of a network of agents that execute a distributed software independently of the blockchain protocol and that limit the execution of the smart contract instructions to only a few steps, necessary to be trustworthy. The implementation of such network consists in the so called "layer two" solution [25], where the same principle of decentralization of DLTs is applied. Indeed, an instance of such layer two solution would be the use of a second DLT with different features in respect to the "main" one [56] where to write the negotiation outcome, e.g. consensus mechanism, or faster operations execution.

A good compromise between the two architectures would be the use of this architecture, i.e. a decentralised oracle, over the others to perform the argumentation, negotiation and interaction with TransactionSC. The data needed to execute these processes, such as the argumentation graph, would be stored in a lower cost secondary DLT or other layer two technology that preserves the immutability of the data. The execution

of the negotiation could take place outside the chain and then be "committed" [25] on the main chain using a hash function to be immutable and therefore verifiable. It would not be susceptible to a single point of failure and the cost of execution overhead would be favourable compared to the second architecture.

Table 1. Comparison between the three architectures considered.

	Argumentation Graph Off-chain	Negotiation Execution Tracing	No Single Point of Failure	No Execution Overhead Costs	Tamper-resistant	Verifiability	Privacy
Architecture 1 **Centralized**	✓	×	×	✓	×	×	✓✓
Architecture 2 **Smart Contract**	×	✓✓	✓	×	✓✓	✓✓	×
Architecture 3 **Decentralized**	✓	✓	✓	×	✓✓	✓✓	✓

5 Intelligent Human-input-based Blockchain Oracle (IHiBO)

In this section, we propose IHiBO, that leverages blockchain and smart contracts framework, which provides a favourable environment with their salient properties, i.e., auditability, traceability and transparency. IHiBO can deal with the potentially inconsistent information input by human experts: we explained how the system may manage the information by argumentation and negotiation considering three possible architectures.

5.1 Combining Formal Argumentation and Negotiation with the Blockchain Framework for Transparency

Argumentation has the ability to provide various ways for explaining why a claim or a decision is made. In this section, the IHiBO we propose might have particular relevance in cases where the decision making process about what data should be fed in the smart contract needs to be transparent: for fund management, the investors don't know what exactly happens to their money, and especially why, so the question whether the fund managers do fulfill their legal and ethical commitment of acting in the best interest of the investor might remain unanswered.

In general, the transparency that can be gained due to the proposed intelligent oracle architecture could be highly valuable in any trust services. The concept of the fiduciary is based on—as the name of these services show—trust: it requires being bound both legally and ethically to operate and use its expertise in the investor's best interests on the

fiduciary's side, and it requires trust on the investor's side to believe in that the fiduciary has done and will do so. This trust can be, to some extent, replaced by intelligent, decentralized solutions providing full transparency of, for instance, fund management: not only the transactions can be fully traced but the expert opinion input and the decision mechanism too. By implementing argumentation and negotiation phases through oracles into smart contract or make them on a side-chain can generate more transparency for investors: investors can know how the final decision is made at the end of reasoning. This could be highly relevant for the investor practicing his right to check the fiduciary's activities in the case of an asset management contract. From Explainable AI perspective, Architecture 2 and Architecture 3 offer an explanation to how a specific decisions has been made.

5.2 Legal Considerations

Next to the technical and financial aspects, legal considerations should also be taken into account when comparing the different architectures. While our motivation is to provide transparency regarding the decision-making process to the principal to gain some insights whether the work of the fiduciary indeed happens according to his best interest, the transparency one should gain with using DLTs is subject to serious limitations.

On one hand, the the principal's right to check is not limitless, it concerns strictly the processes of managing his assets, but more importantly, given the characteristics of DLTs, a(n unwantedly) broader audience would be involved in the disclosure of information if one chose not the appropriate architecture, threatening trade secrets and involving privacy problems.

On the other hand, once the application of DLTs become widespread in the securities market, mandatory disclosure rules motivated by anti-tax avoidance should be aligned with the new technology [61]. Indeed, DLT-based automated disclosure may lead to the release of information that is too fast, limiting the ability of investors to properly speculating. Thus, mandatory disclosure requirements would still be necessary, but the enforcement of such provisions and detection of violations redesigned using DLTs and smart contract would have to deal with the necessity of stakeholders.

Architecture 3 seems to be the best option from these point of view too: in contrast to the public, permissionless verification that DLTs usually employ while smart contracts are executed, layer two solutions usually move this process off-chain. This definitely poses security issues compared to a protocol executed completely on-chain, however there are currently some viable solutions proposed that address this issue [24]. For instance, an application might be the use of a permissioned sidechain. In this case, information that would clash with trade secrets and privacy would be stored on that permissioned chain and maintained by the participants who have been nominated for this, e.g. joint data controllers as permissioned blockchain operators [37]. Through the use of commitments on the main chain [25], i.e., the permissionless one, the necessary steps for verification are implemented, and once the fiduciaries operating the sidechain reveal part of the information to the principals, the latter can verify its validity on-chain [56].

5.3 IHiBO Direction

We argue that a layer two solution, the decentralized oracle solution in Architecture 3, provides the proper mid ground in terms of cost of execution, for latency and fees, and verifiability of the complete process. Indeed, there might be use cases where some data should not be disclosed, and an argumentation and negotiation architecture based on a full execution on smart contracts would not allow it. In the other extreme case, for a centralized oracle, the entire process behind a decision made could be concealed or its log could be altered. In a decentralized oracle architecture the complete execution could be logged off-chain and then committed on-chain, making it impossible to alter the logs, while not disclosing these entirely [25]. Members of the management body[8] shall have adequate access to information and documents which are needed to oversee and monitor management decision-making[9]. In our second and third architectures, each execution of all the smart contracts can be audited, validated and maintained by every participants, thus reduce the time and fee of extra work of surveillance, which will in turn reduce potential corruption or conflicts of interests.

6 Discussion

IHiBO can develop the degree of trust in several ways. As argued by Walton, it seems to be more generally acknowledged now that we do have to rely on experts, and that such sources of evidence should be given at least some weight in deciding what to do in practical matters [65]. In our case study, managers play the role of experts and the professional certificate of them as well as their past creditable experience could be part of the backup of trustworthiness of the source information, and we calculate the weight of the arguments in the parallel of voting theory, i.e. to count the number of supporting managers. Formal argumentation systems have been computationally implemented that can be used to model arguments from expert opinion and to evaluate them when they are nested within related arguments in a larger body of evidence. One such system is ASPIC+ [49]. ASPIC+ is based on a Dung-style abstract argumentation framework that determines the success of argument attacks and that compares conflicts in arguments at the points where they conflict [21]. Our adoption of agent argumentation is also originated from Dung's framework, while we extend it with the role of agents and associated relation with arguments. Such that together with blockchain technology, investors are in a clear position to audit the source of arguments, and the way they communicated.

Another way to gain and restore trust from investors is to make the resources and decision-making process explicit, our case can be considered as a good example of the use of argumentation for favouring trust. Being skillful and sophisticated could be

[8] Art. 4(8) MiFID II: 'management body' means the body or bodies of an investment firm, market operator or data reporting services provider, which are appointed in accordance with national law, which oversee and monitor management decision-making and include persons who effectively direct the business of the entity.

[9] DIRECTIVE 2014/65/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 15 May 2014 on markets in financial instruments and amending Directive 2002/92/EC and Directive 2011/61/EU

not enough for the requirement of managers. Especially when they are in corporation, other problems may arise to obtaining trust, like reliability and agency problems. For instance, problems arising from managers' unwillingness and lack of incentives to act in the principal's best interests, rather than from a lack of expertise. In our case design, investors have the advantages to audit the resources of the information, thus such risks could be mitigated.

Falcone and Castelfranchi relate trust explicitly to the goals of agents, and consider trust to be concerned with whether another agent can and will perform an action that will enable the first agent to achieve its goals [22]. In the case of fund management, fund managers are sharing the same goal——gain interests for the investors. In the study case, agents must coordinate and communicate with their own information to reach an agreement. In this scenario, the requirement to reach trust is to ensure and audit the trustworthiness of a source of information within an argument which is then to be decided to be accepted or not. We ensure the trustworthiness of the information by counting the values or the numbers of support from agent to arguments to ensure the resources based on somehow voting theory.

On the other hand, the adoption of blockchain and DLT has been under consideration for several years both from economic and legal aspects [26,51]. However, most of them only consider the transaction process, i.e. how to use these technologies for clearing and settlement, and some propose to use smart contracts to conduct the functions of CCP or central securities depository (CSD) [10] [44]. In our work, we pay attention to the pre-trading phase, where the investment decisions made by the trust services are extremely crucial to investors. As discussed above, the decision-making process is traceable and immutable on blockchain. As a result, the entire reasoning decision and transaction process are transparent and investors can gain maximum confidence and thus trust for the trust services.

7 Related work

Our methodology is a hybrid of decision-making based on formal argumentation, autonomous negotiation, blockchain, smart contracts, and oracles, all of these are serving for the trust service, thus, we need to look at the related work from multiple perspectives. To the best of our knowledge, there is no mature work on adopting argumentation in the financial world. The only work we can find is to use argumentation as a convincing tool in order to gain the stakeholders' support and trust; it also mentions that argumentation is a communicative interaction which conducts the claims as propositions, e.g. "You should invest in Treasury Bonds" [46].

There is influential work on argumentation and trust has been done. First of all, trust in information sources has been used in argumentative reasoning. This is also true with respect to the exchange of arguments in social interaction. When people argue with other parties, trying to make their arguments accepted to reach a final agreement,

[10] CSDs operate the infrastructure that enables the securities settlement, allow the registration and safekeeping of securities, allow the settlement of securities in exchange for cash, track how many securities have been issued and by whom, track each change in the ownership of these securities

they also evaluate the arguments proposed by the opponents in the discussion. In the earlier work on argumentation theory, people only focus on the relation among arguments, i.e. the arguments are considered to be accepted or not depending on the attacks against them [53]. Neither the information sources nor their trustworthiness degree are considered. In recent years, the area has seen a number of proposals [38,48,57,60,64] to introduce the trust component in the evaluation process of arguments. Argumentation also has been used to reason about trust evaluations. Trust is a process of critical judgement rather than a blind altitude where argumentation can come into play as a powerful tool to reason about trust, making sure such trust is well-built. In their earlier work, Parsons et al suggest argumentation might play a role which tracks the origin of information used in reasoning, thus it can provide provenance in trust [48]. Later the same authors develop a general system of argumentation that can represent trust information, and be used in combination with a trust network, using the trustworthiness of the information sources as a measure of the probability that information is true [59].

In the IHiBO architecture, we use the oracles requiring input which involves human intervention. Human oracles are rarely applied [18]. The rare existing ones are applied in applications with binary inputs, i.e., they only take input by one of two possibilities, typically ”yes” or ”no” [43]. This greatly narrows the scope of questions the answer to which we could rely on human experts. There can be cases where the missing input is not binary, but contains further and different types of data, while the generation of input of some smart contracts requires in particular human subjective judgment. The advantage of human assessment is also apparent in situations where contractual performance must be evaluated holistically, rather than by simple measurement of specific parameters.

8 Conclusion and Future Perspectives

The main contribution of this paper is proposing an integrated framework which incorporates formal argumentation and negotiation within a blockchain. These techniques have distinctive features that complement each other. They together make the decision-making processes of fund management transparent and traceable. As a result, our methodology enhances trust from principals to trust services, especially the famous form of trust, i.e., knowledge-based trust [34], which is grounded when knowing the other (fund management) sufficiently well so that the behavior of managers can be understood and predicted more accurately. Our motivation came from trust services, so we explained our idea in a fund management scenario, but our proposal is not bound to this domain.

One follow-up possible work is to provide and adapt to a high level of adaptability in the decisions of the fund management. For instance, to define different investment scenarios according to the investors' preferences, attitude (aggressive or moderate) and the financial environment (e.g. bull or bear market), including the possibility to forecast the status of the financial market for the next investment period, in order to select the ones which will bring the biggest interests. Besides, we plan to explore the combination of negotiation and argumentation. For instance, here we adopt a simplified example on fund investment, the real life relying on existing works proposing argumentation-based negotiation is a useful next step since exchanging justified information among agents

gives them enough knowledge to try and reach a common understanding much faster [12].

Another possible work could be to investigate the integration of consensus mechanisms for a layer two solution to the dispute resolution phase, in order to narrow the gap between blockchain and argumentation as well as negotiation, since there is no specialized blockchain yet that has a protocol that integrates reasoning. For instance, if there is a blockchain based on *Proof of Stake* (instead of *Proof of Work*), validators need to vote to validate a transaction based on a reasoning process where each validator has a different set of knowledge data.

Lastly, we also plan to rely on the recent advances of the domain of Explainable AI [4] to explore how we can make the decision-making process presented in this paper explainable for different types of users (experts, non-experts, etc.) and for different purposes (e.g. transparency, debugging, etc.).

Acknowledgements

This work has received funding from the EU H2020 research and innovation programme under the Marie Skłodowska-Curie Actions Innovative Training Networks European Joint Doctorate grant agreement No 814177 Law, Science and Technology Joint Doctorate - Rights of Internet of Everything.

References

1. Leila Amgoud and Claudette Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning*, 29(2):125–169, 2002.
2. Leila Amgoud and Claudette Cayrol. On the acceptability of arguments in preference-based argumentation. *CoRR*, abs/1301.7358, 2013.
3. Leila Amgoud and Srdjan Vesic. Rich preference-based argumentation frameworks. *International Journal of Approximate Reasoning*, 55(2):585–606, 2014.
4. Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. Explainable agents and robots: Results from a systematic literature review. In *18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, pages 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems, 2019.
5. Katie Atkinson, Pietro Baroni, Massimiliano Giacomin, Anthony Hunter, Henry Prakken, Chris Reed, Guillermo Simari, Matthias Thimm, and Serena Villata. Towards artificial argumentation. *AI magazine*, 38(3):25–36, 2017.
6. Tim Baarslag, Koen Hindriks, and Catholijn Jonker. Acceptance conditions in automated negotiation. In *Complex Automated Negotiations: Theories, Models, and Software Competitions*, pages 95–111. Springer, 2013.
7. Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10-15):675–700, 2007.
8. Moritz Becker and Balázs Bodó. Trust in blockchain-based systems. *Internet Policy Review*, 10(2), 2021.
9. Abdeljalil Beniiche. A study of blockchain oracles. *arXiv preprint arXiv:2004.07140*, 2020.

10. Guido Boella, Souhila Kaci, and Leendert Van Der Torre. Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 107–118. Springer, 2009.
11. Paul M Bosse, Douglas M Grim, and CFA Frank Chism. Duty, opportunity, mastery: Investment committee best practices, 2017.
12. Luís Brito, Paulo Novais, and José Neves. The logic behind negotiation: from pre-argument reasoning to argument-based negotiation. In *Intelligent agent software engineering*, pages 137–159. IGI Global, 2003.
13. Vitalik Buterin et al. Ethereum white paper, 2013.
14. Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 378–389. Springer, 2005.
15. James Chen. Fund manager, 2021.
16. Robert G Cooper, Scott J Edgett, and Elko J Kleinschmidt. Portfolio management in new product development: Lessons from the leaders—i. *Research-Technology Management*, 40(5):16–28, 1997.
17. Robert Gravlin Cooper. *Winning at new products*. Addison-Wesley Reading, MA, 1986.
18. Matija Damjan. The interface between blockchain and the real world. *Ragion pratica*, pages 379–406, 2018.
19. Sylvie Doutre and Jean-Guy Mailly. Constraints and changes: A survey of abstract argumentation dynamics. *Argument & Computation*, 9(3):223–248, 2018.
20. Phan M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
21. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.
22. Rino Falcone and Cristiano Castelfranchi. Social trust: A cognitive approach. In *Trust and deception in virtual societies*, pages 55–90. Springer, 2001.
23. Peyman Faratin, Carles Sierra, and Nick R Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.
24. Sara Feenan, Daniel Heller, Alexander Lipton, Massimo Morini, Rhomaios Ram, Robert Sams, Tim Swanson, Stanley Yong, and Diana Barrero Zalles. Decentralized financial market infrastructures. *The Journal of FinTech, Forthcoming*, 2020.
25. Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais. Sok: Layer-two blockchain protocols. In *International Conference on Financial Cryptography and Data Security*, pages 201–226. Springer, 2020.
26. Sebastiaan Niels Hooghiemstra. Distributed ledger technology ('dlt') and its impact (on the regulation of) european investment funds. *Available at SSRN 3735886*, 2020.
27. Anthony Hunter and Matthias Thimm. Probabilistic reasoning with abstract argumentation frameworks. *Journal of Artificial Intelligence Research*, 59:565–611, 2017.
28. Nicholas R Jennings, Peyman Faratin, Alessio R Lomuscio, Simon Parsons, Carles Sierra, and Michael Wooldridge. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.
29. Johannes Rude Jensen, Victor von Wachter, and Omri Ross. An introduction to decentralized finance (defi). *Complex Systems Informatics and Modeling Quarterly*, (26):46–54, 2021.
30. Souhila Kaci and Leendert van der Torre. Preference-based argumentation: Arguments supporting multiple values. *International Journal of Approximate Reasoning*, 48(3):730–751, 2008.

31. Andrew Koster, Jordi Sabater-Mir, and Marco Schorlemmer. Argumentation and trust. In *Agreement Technologies*, pages 441–451. Springer, 2013.
32. Yeşem Kurt Peker, Xavier Rodriguez, James Ericsson, Suk Jin Lee, and Alfredo J Perez. A cost analysis of internet of things sensor data storage on blockchain via smart contracts. *Electronics*, 9(2):244, 2020.
33. Olga Labazova. Towards a framework for evaluation of blockchain implementations. 2019.
34. Roy J Lewicki and Barbara Benedict Bunker. Developing and maintaining trust in working relationships.‖ in rm kramer and tr tyler (eds.), trust in organizations: Frontiers of theory and research, 1996.
35. Hengfei Li, Nir Oren, and Timothy J Norman. Probabilistic argumentation frameworks. In *International Workshop on Theorie and Applications of Formal Argumentation*, pages 1–16. Springer, 2011.
36. Beishui Liao, Li Jin, and Robert C Koons. Dynamics of argumentation systems: A division-based method. *Artificial Intelligence*, 175(11):1790–1814, 2011.
37. Tom Lyons, L Courcelas, and K Timsit. Blockchain and the gdpr. In *The European Union Blockchain Observatory and Forum*, 2018.
38. Paul-Amaury Matt, Maxime Morge, and Francesca Toni. Combining statistics and arguments to compute trust. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 209–216. Citeseer, 2010.
39. Massimo Morini. Managing derivatives on a blockchain. a financial market professional implementation. *A Financial Market Professional Implementation (May 5, 2017)*, 2017.
40. A Najjar. *Multi-agent negotiation for qoe-aware cloud elasticity management*. PhD thesis, PhD thesis, École nationale supérieure des mines de Saint-Étienne, 2015.
41. Amro Najjar, Xavier Serpaggi, Christophe Gravier, and Olivier Boissier. Multi-agent negotiation for user-centric elasticity management in the cloud. In *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, pages 357–362. IEEE, 2013.
42. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
43. Keerthi Nelaturu, John Adler, Marco Merlini, Ryan Berryhill, Neil Veira, Zissis Poulos, and Andreas Veneris. On public crowdsource-based mechanisms for a decentralized blockchain oracle. *IEEE Transactions on Engineering Management*, 67(4):1444–1458, 2020.
44. Simona-Vasilica Oprea, Adela Bâra, and Anca Ioana Andreescu. Two novel blockchain-based market settlement mechanisms embedded into smart contracts for securely trading renewable energy. *IEEE Access*, 8:212548–212556, 2020.
45. Fabio Paglieri. Trust, argumentation and technology. *Argument Comput.*, 5(2-3):119–122, 2014.
46. Rudy Palmieri. Regaining trust through argumentation in the context of the current financial-economic crisis. *Studies in Communication Sciences*, 9(2):59–78, 2009.
47. Wenxuan Pan and Meikang Qiu. Application of blockchain in asset-backed securitization. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pages 71–76. IEEE, 2020.
48. Simon Parsons, Peter McBurney, and Elizabeth Sklar. Reasoning about trust using argumentation: A position paper. In *International Workshop on Argumentation in Multi-Agent Systems*, pages 159–170. Springer, 2010.
49. Henry Prakken. An overview of formal models of argumentation and their application in philosophy. *Studies in logic*, 4(1):65–86, 2011.
50. Henry Prakken and Giovanni Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of applied non-classical logics*, 7(1-2):25–75, 1997.
51. Randy Priem. Distributed ledger technology for securities clearing and settlement: benefits, risks, and regulatory implications. *Financial Innovation*, 6(1):1–25, 2020.

52. Dean G Pruitt. *Negotiation behavior*. Academic Press, 2013.
53. Iyad Rahwan and Guillermo R Simari. *Argumentation in artificial intelligence*, volume 47. Springer, 2009.
54. Howard Raiffa. *Negotiation analysis: The science and art of collaborative decision making*. Harvard University Press, 2007.
55. Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, pages 97–109, 1982.
56. Amritraj Singh, Kelly Click, Reza M Parizi, Qi Zhang, Ali Dehghantanha, and Kim-Kwang Raymond Choo. Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *Journal of Network and Computer Applications*, 149:102471, 2020.
57. Ruben Stranders, Mathijs de Weerdt, and Cees Witteveen. Fuzzy argumentation for trust. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 214–230. Springer, 2007.
58. Katia P Sycara. Multiagent systems. *AI magazine*, 19(2):79–79, 1998.
59. Yuqing Tang, Kai Cai, Elizabeth Sklar, Peter McBurney, and Simon Parsons. A system of argumentation for reasoning about trust. In *Proceedings of the 8th European Workshop on Multi-Agent Systems, Paris, France*, 2010.
60. WT Luke Teacy, Jigar Patel, Nicholas R Jennings, and Michael Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.
61. Muthukkumarasamy Thuvarakan. Regulatory changes for redesigned securities markets with distributed ledger technology. *The Knowledge Engineering Review*, 35, 2020.
62. Rijo Jackson Tom, Suresh Sankaranarayanan, and Joel JPC Rodrigues. Agent negotiation in an iot-fog based power distribution system for demand reduction. *Sustainable Energy Technologies and Assessments*, 38:100653, 2020.
63. Leon van der Torre and Srdjan Vesic. The principle-based approach to abstract argumentation semantics. *FLAP*, 4(8), 2017.
64. Serena Villata, Guido Boella, Dov M Gabbay, and Leendert Van Der Torre. Arguing about the trustworthiness of the information sources. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 74–85. Springer, 2011.
65. Douglas Walton. On a razor's edge: Evaluating arguments from expert opinion. *Argument & computation*, 5(2-3):139–159, 2014.
66. Gerhard Weiss. *Multiagent Systems*. MIT Press, 2013.
67. Michael P Wellman, Amy Greenwald, and Peter Stone. *Autonomous bidding agents: Strategies and lessons from the trading agent competition*. Mit Press, 2007.
68. Sam M Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William J Knottenbelt. Sok: Decentralized finance (defi). *arXiv preprint arXiv:2101.08778*, 2021.
69. Michael Wooldridge. *An introduction to multiagent systems*. John wiley & sons, 2009.
70. Liuwen Yu, Dongheng Chen, Lisha Qiao, Yiqi Shen, and Leendert van der Torre. A principle-based analysis of abstract agent argumentation semantics. 2021. under review.
71. Liuwen Yu, Réka Markovich, and Leendert Van Der Torre. Interpretations of support among arguments. In *Legal Knowledge and Information Systems*, pages 194–203. IOS Press, 2020.
72. Liuwen Yu and Leendert van der Torre. A principle-based approach to bipolar argumentation. In *NMR 2020 Workshop Notes*, page 227, 2020.
73. Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Weili Chen, Xiangping Chen, Jian Weng, and Muhammad Imran. An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105:475–491, 2020.

74. Mirko Zichichi, Stefano Ferretti, Gabriele D'Angelo, and Víctor Rodríguez-Doncel. Personal data access control through distributed authorization. In *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pages 1–4. IEEE, 2020.
75. Mirko Zichichi, Stefano Ferretti, and Gabriele D'angelo. A framework based on distributed ledger technologies for data management and services in intelligent transportation systems. *IEEE Access*, 8:100384–100402, 2020.

Social robotics and deception: beyond the ethical approach

Rachele Carli[1][0000−0002−8689−285X]

1. Alma Mater Research Institute for Human-Centered AI, University of Bologna, Italy; 2. ICR group, University of Luxembourg, Luxembourg
rachele.carli2@unibo.it

Abstract. Social robots are designed to directly interact with users, to collaborate with them and to act in a human-centred environment, with different degrees of automation. In order to encourage acceptability and trust, they are structured as so to leverage the human tendency to anthropomorphise what they interact with. It follows that some machines are able to simulate the feeling of genuine emotions or empathy, to appear needy of help, to pretend to have an own personality and – more in general – to induce the user to think that they are something more than mere objects. Thus, it may be argued that such interaction could lead to forms of manipulation that fall within the remit of a deceptive dynamic. Such a phenomenon is still much debated by the scientific community and raises significant concerns regarding long-term ethical and psychological repercussions on the users.
This paper investigates which tools we have and which ones we may need to tackle the theme of deception in social robotics. Therefore, both ethical and legal perspectives are reconstructed, with the attempt to try to distinguish their respective scope and to emphasise their fruitful integration in addressing these issues. Finally, the possible relevance of fundamental human rights in human-robot interaction dynamics is discussed, due to their ability to reconcile ethical demands with the binding feature of legal norms.

Keywords: HRI · Deception · Human Dignity · Ethics · Law

1 Introduction

The so called "Fourth Revolution" [24] is leading to the development of new technological devices, increasingly interactive and pervasive in many areas of our lives. We are assisting to the migration of robots from factories to our homes, involved in many tasks - ranging from education to health, from entertainment to the care of the most fragile ones - [12]. This explains the growing focus on social robotics. In fact, social robots are characterised by a software, which allows them – based on the level of technological advancement – to perceive the environment, to interpret both structured and not structured data, to process them and to extrapolate primitive and derivative pieces of information [27]. Therefore, they are able to directly interact with the users, collaborating with them on a daily

base and performing multiple tasks with different degree of automation. Some of them are also able to develop social competences, create social bonds and learn to appropriately use natural signals – like indicating, gazing, winking – [14].

Consolidate scientific literature demonstrates that facilitating acceptability and trust in social robots plays a central role in order for them to pursue the given tasks and to behave efficiently in a human-centered environment [84][72]. Therefore, they are designed and programmed so as to lever the human tendency to attribute anthropomorphic characteristics to what they relate to, repeatedly over time [89].

This might lead to a new form of manipulation, based on deception. Although the theme of "deception in social robotics" is becoming a reason for multidisciplinary debate, there is still no unequivocal definition of this concept when used to define a dynamic of human-machine interaction. For the purposes of this analysis we will consider the cases in which the characteristics of the robot may alter the user's perception of the capabilities it has or does not have. In fact, in spite of the pleasant design, the ability to move and act like living beings, to simulate pertinent conversation and to emulate feelings and emotions, robots are just objects [5]. They are not capable to set autonomously a goal and even the most sophisticated functionality is - at least for the moment - the result of the way they have been programmed by a human expert. Such a deceptive dynamic, although not favoured with dishonest intents, may be potentially risky for the physical, economical and psychological integrity of people involved and for the authenticity of their will. Moreover, it should be underlined that the speed of development of these technologies far exceeds the speed with which we are able to investigate negative outcomes. On one hand, possible side-effects - especially on the ethical, psychological and sociological level - have already been theorised. On the other hand, not all of them can be already precisely measurable or unequivocally proved, for we cannot have a long-term picture of the expected consequences yet. To this end, both the ethical and the legal perspectives play a central role. However, the respective scopes of intervention must be identified, in order to guarantee the efficacy of their impact on the theme of deception in social robotics.

This paper aims to investigate the effectiveness of ethics and the law as tools of analysis and to suggest the relevance of fundamental human rights – in particular human dignity – as a valid option to tackle this theme. In fact, they have the advantage to be efficiently used in a multidisciplinary debate, increasing the opportunities to find a common solution for similar cases [79] and to suggest a transparent frame of regulation. This would be functional to guarantee that new technologies (i) are projected so as to respect the centrality of human being, (ii) can be efficiently tested in the real world and (iii) are commercialised in a specifically developed market [59].

To this end, Section 2 recollects some relevant literature about the theme of deception in the human-robot interaction context, in order to give a general understanding of the phenomenon. Section 3 briefly presents some scenarios in which deception - theorised as in the previous section - can occur. The attention

will be focused on two categories of users, children Subsection 3.1 and fragile people Subsection 3.2, trying to highlight which concerns may rise from such a dynamic. Section 4 evaluates ethics and the law as possible tools to address this theme. In particular, some of their strengths and weaknesses will be separately analysed in Subsections 4.1 and 4.2. Therefore, Section 5 suggests that human dignity may play a more effective role as balancing principle with a view to risk-benefit examination of social robotics and the deceptive phenomenon.

2 Main forms of deception in human-robot interaction

The ability of a machine to deceive the human counterpart has been considered the qualifying criterion for the very notion of AI. This is due to the well-known "Turing Test", according to which a machine would have been considered as "intelligent" if it was able to induce a person – placed in a different room – to believe to be chatting with another human being, rather than with a robot [47]. However, there is still not a univocal understanding and evaluation of what is meant for deception in the context of human-robot interaction. This is due to the fact that simulation mechanisms depend closely on the nature of the robot, its functionality, the tasks it has to perform, the object of the interaction [71].

In order to better analyse the phenomenon, it is useful to introduce a preliminary tripartition: (i) external state deception, (ii) superficial state deception, (iii) hidden state deception [13].

The first one takes place (i) when the robot lies about something regarding the external world. It could be considered problematic, when it aims to mislead the user, but it can also imply a pro-social function. In fact, social conventions can require "white lies" for several reasons, such as to carry on the conversation, to be polite, to avoid uncomfortable situations or to encounter one's favour. This is possible using hyperbole or pleasantries, improvising not really known pieces of information and managing expectations [87]. Implementing machines with such features means favouring their integration into the human environment and overcoming the prejudices in which they may incur [38].

More challenging is the case of robots which (ii) simulate to possess capabilities and emotional dimensions actually lacking. With regards to this aspect, there are different opinions among the experts. According to the most extreme position any robotic cue that emulates a typically human one is deceitful [86]. This on the base of both technical and philosophical assumptions. On one hand, the behaviour of a device is evaluated as nothing more than the result of the way it was projected by the programmer [23], being neither aware nor autonomously settled. On the other hand – following the same argumentative line – anything is able to manipulate reality should be considered ethically wrong, for it harms the "duty to see the world as it is" [77]. A more lenient position traces back to the category of deception only those actions that induce the user to perceive the machine as something more than a very sophisticated piece of equipment [31], closer to a living being [76]. Such a function is conveyed not only by gestures or movements, but even more so by the simulation of emotional capabilities [49],

the emulation of feelings of pain, suffer, attachment, care [39]. This is certainly functional for the collaboration between the user and the artefact. Nonetheless, it can also affect subconscious social dynamics, interfering with the formation and expression of the individual's will. Moreover, it is important to underline that such characteristics have more incisiveness on lonely and needy people, the same ones that should be better protected against manipulative mechanisms, for more vulnerable.

The third form of deception occurs when the robot (iii) takes advantage of emulative signals in order to hide capabilities it has. This can lead to harmful consequences for people's privacy and data managing. For example, it was demonstrated that children and the elderly are more likely to confide to a "friendly" robot even things that they would not have revealed otherwise. This because they are persuaded that the machine can keep the secret and because they are unaware of how it can process those confidences in order to target their desires and preferences [68]. In fact, in order to improve both engagement and quality of the interaction, social devices can record actions, words and even emotions [48]. Furthermore, an individual could believe that when the robot is not in the view it cannot record what the person is doing [40], ignoring the presence of sensors that make him/her lives in a sort of "Big Brother" [56]. The lack of full understanding of the effective functionality of the machine could undermine the value of the consent – no more considerable as "informed" – given to the interaction [83]. At the same time, it is not objectively demonstrable that an increase in the information provided will always lead to a greater awareness in the use of the device. In fact, human-robot interaction involves multiple factors, many of which are strictly related to a subjective psychological dimension [8].

Taking into consideration the above-mentioned classification of the main forms of deception in the human-robot context, it is relevant to analyse some concrete scenarios in which such a dynamic may occur. Thus, potential long-term consequences may be analysed, so as to distinguish beneficial and harmful effects. This evaluation is fundamental, in light of the necessity of a human-centred development of AI systems.

3 Possible deceptive scenarios

It was assumed that social robots leverage the natural human tendency to anthropomorphising inanimate things (section 1). Indeed, Freud defined humans as "symbolic animals" [9], who tend to create and modify the way in which reality appears to them. Therefore, someone could argue that the individuals involved in the interaction have their own responsibility in the process of deception. In such a view, the machine's deceitful behave would appear as less relevant.

Nevertheless, it is important to introduce a fundamental distinction between two terms: (1) anthropomorphism and (2) anthropomorphisation. While the first one refers to the human propensity to attribute human-like features to robots [34], the second one implies the deliberate choice of designing such characteristics by developers [53]. It follows that, even if the attribution of anthropomorphic

traits to the machine had not the precise aim to deceive – or to do so with a malicious intent – the programmers would have had the competence to foresee this effect and to correct its potentially harmful drift [70]. In fact, fully rational people are subjected to such a dynamic too [8]. It could be objected that we are not facing real deception. In fact, there are other circumstances in which people are entertained through an illusion and still maintain the ability to distinguish it from reality, without negative consequences. Those who support this idea make the example of a magic show [11]. As the spectators know that the magician does not actually cut the partner into two parts - and are amused instead of scared -, in the same way robots' users could know that the machine is simulating emotions and attachment, without really experiencing them.

Though, the characteristics which influence humans in anthropomorphising these devices are concretely present by design. For this reason, it was underlined that the difference between a mere toy - for instance - and a robot is the same as between the action of pretending and that of believing [81]. An example is the one of robots deliberately structured so as to seem clumsy, in need of help, or to make mistakes in pursuing the given task. This design choice is due to the fact that error is typically considered as human-like, while efficiency and perfection of execution are usually linked to what is artificial [64]. Thus, the creation of an empathic bond is elicited. Likewise, it was demonstrated that a similar effect can be produced by implementing the machine with a 'cheating' functionality [73]. The result is maximised if the robot repeatedly deceives the user, for it encourages the perception of an autonomous will in the device [46]. On the contrary, if the cheating behaviour is carried out only once, it is more likely considered as a problem of malfunction. This underlines that, in the context of a human-social robot interaction, efficiency in operation is expendable in favour of the possibility of living a human-like experience.

Therefore, evoking a social behavior from the user is the result of a deliberate choice, not just an accident due to the nature of the subject of the interaction. On the contrary, human nature and inclination has been accurately investigated so as to make it easier to cause a particular belief and attitude towards the technical device.

When the interaction involves subjects that are more vulnerable due to their age of health condition, the effects of continued exposure to similar mechanisms deserve to be analyzed more in detail. Consequently, plausible scenarios of robots with children and with fragile people will be presented below.

3.1 Children-robot interaction

The illusion that machines can be engaged in an appropriate conversation, feel empathy and establish a real friendship can lead to entrusting them with tasks that go far beyond their actual functionalities. In order to better understand this passage, we can try to image a children's play scenario. The robot can be trained to prevent or to react to standard/common hypothesis of harmful events. However, it has very little changes to recognise a child pretending to fall – because of the nature of the play – from one who has actually been hit. Again,

a child who uses a common tool – such as a pen or scissors – does not always use it appropriately and the machine may not be able to distinguish – or distinguish promptly – the suitable use from the harmful one [70].

With regard to childhood, then, there is a heated debate about the possible uses of social robots. Some studies show that these devices have a positive effect in the treatment of children with autism [67][17][37]. However, it may be the case that what was presented as a solution at the very beginning may turn to be the problem at the end. Due to the mechanical, precisely planned nature of the machine a long-term/semi-exclusive interaction with people tending to isolation could increase this practice. In fact, the machine represents a 'safe reference', which does not pose opposition and promptly meets the unidirectional needs of the child [15].

The most detrimental effect of such a deprivation of significant human relationships could be appreciated in babies. In this case, we can refer to indirect evidence only, for it is not admissible to conduct experimentation with newborns. Old researches highlight that those who were deprived of a 'personalised', attentive, empathic care or of warmth, human contact let themselves die or developed serious physical and psychological problems [54][10].

Even if we do not consider babies, but older children, a long-term interaction with robotic caregivers, instead of human ones, could imply criticality. First of all, they will become used to predictable and schematic responses to given inputs, possibly developing difficulties in managing real emotions – like disappointment, frustration, dissatisfaction –. This could compromise their capacity to empathy themselves, for they would lack of the experience of real relationships, based on compromise, mutual-adaptation, in which it is impossible to be always listened, pleased and pandered in selfish needs [82].

Nevertheless, it cannot be ignored that some applications of AI devices with young people have also positive aspects. This is the example of Nao or Pepper, which can help children manage painful medical procedures or not be completely excluded from the school context due to a long hospital stay [57]. It follows that to accurately ponder the kind, time and dynamic of the interaction can be fundamental in order to distinguish empowering uses of social robots from detrimental ones.

Moreover, it was demonstrated that, when the robot simulates gratitude or a more intimate interaction with a specific child because of the amount of attention he/she turned on it, the child was encouraged to increase this behaviour [41]. In fact, the more the user interacts with the machine the more the result will be satisfactory and calibrated on the personality and the habits of the human being [36]. In addition, the overexposure to technological devices has been proven to release dopamine and its sudden deprivation or decrease can provoke anxiety, restlessness, anger . So described, the pattern is close to the one established in case of any form of addiction - both behavioural and substance ones –.

In the case of robots that engage the user at an emotional level, these effects can be summed to those of psychological attachment and affective dependence. Though, such a long-term result should be deepened with specific studies.

Indeed, it should be underlined that new technologies have an impact on how we act in the world, being able to modify the way we perceive and concept reality . This is even more the case for the youngest ones, who have not completed their psychological and cognitive development path yet.

3.2 Social robots and fragile people

Social robots can be used even in the treatment of people with mental or physical disability and the elderly. In these scenarios, the device can have the role of a caregiver, a companion or even a therapeutic tool. It is easily understandable that the machine's deceptive features have different effects on the base of the task it has to perform.

An example is Paro [62], a device resembling a seal pup which displays positive responsiveness and beneficial impact on the health and mental status of the user when cuddled [61]. The choice to emulate this specific animal is not by chance. It is certainly not a typical pet – such as a cat or a dog – and this decreases human's expectations with regard to the way it responds to the interaction [30][66]. Thus, its technology is able to influence people's emotions, although it is not very sophisticated. Another case is the one of a robotic doll, specifically programmed to induce individuals affected by dementia to create an emotional bond towards the machine [43], engaging them at a conversational level. A similar dynamic was analysed in the project Rehabibotics, involving people with serious cognitive difficulties. The robot was projected so as to provoke empathic responses in order to favour the interaction. Therefore, it could record likes and dislikes of the patients, emotional and mental states, in order to predict them and track the progress of the disease. Anyway, the machine showed some errors in this procedure, which could not be corrected by individuals' feedback, for dementia made them not always – or not reliably – aware of their own inner states [74].

In particular for what fragile and old people are concerned, exacerbation of isolation end dehumanisation [69] are the main risks that need to be carefully considered, for they can lead to the objectification of human beings, whose autonomy and self-determination could be challenged [55].

Moreover, the report written by the Rathenau Institute for the Council of Europe highlights a possible infringement of fundamental rights - in particular human dignity – by the long-term exposure to a continued human-robot interaction for the elderly. This rises the necessity of a reflection with regard to a plausible right to meaningful human contact [20].

It follows that – as we have briefly tried to demonstrate here – social robotics is a varied field, which lends itself to many possible applications. Therefore, the challenge that new technologies poses to social sciences is to identify intervention tools capable of protecting the integrity of the human beings involved in the interaction, taking into account possible material – but also immaterial – damages [1]. To this end, both legal and ethical approaches should be analysed.

4 Ethics and the law: possible tools of analysis

Assessing the theme of deception in social robotics, philosophical and legal perspectives will be taken into account in this discussion. In fact, both of them could be relevant to discuss possible harmful repercussions on the individuals involved and to intervene in order to limit or remove them.

Certainly, they are not the only two tools considered in the investigation of new technologies. However, law and ethics are very often presented as opposing disciplines and the role that they can have in the evaluation and regulation of robotics and AI are often based on conceptual and theoretical misunderstandings. Therefore, identifying general characteristics of each of the two fields is essential to understand how and to what extent they can effectively contribute to the debate.

4.1 The not-universality and not-univocality of ethical statements

By definition, ethics is a branch of philosophy which guides people's behavior in the world and in the relations that they establish one another [25]. For this reason, someone says that whenever there is a debate regarding which conduct or risk is best to take, it has to be ethical oriented [75].

This very approach has been largely adopted even with regards to new technologies [2][26]. However, this discipline has no external oversight nor even standards protocol for enforcing its guidelines [51]. Moreover, it is far from being really universal, contrary to what it is commonly claimed. The term 'ethics', without any other specification, includes different theoretical frameworks of reference and not all of them can be considered conform to every legal system. An example can be the concept of development formulated by transhumanists. Taking into consideration the European context, it appears simplistic and potentially dangerous for the integrity of the users. In fact, it aims to subvert the very concept of "humanity", in favour of a limitless trust in the power of science [78]. Even more radical is the post-humanist refusal to adopt an anthropocentric perspective [28]. The base of this idea is the belief that human nature would be something to be overcome in order to realise "singularity" [80]. With a view of consistency with Member States' Constitutions and international treaties, the bio-conservative understanding of human nature seems the most appropriate to address the issues posed by robotics and AI. It is considered as universally recognized to everyone in reason of their own existence, not modular or subjected to renunciation [42]. However, not even in a similar prospective all the alternatives may be equally suitable. This is the case of utilitarian argumentation. It aims to legitimate deception in human-robot interaction in reason of the beneficial effects that it allows to pursue, without taking into account the wider range of interests and rights involved and the correlate effects [6].

This is possible for ethics purports to investigate all areas of what is rationally knowable, without being held to strictly comply with acquired concepts and axioms – contrary to what concerns the legal analysis –. Therefore, its assumptions and guidelines have been accused of ambiguity and not being obligatory

[85] [50]. Therefore, the variety of existing ethics, the need of a careful *ex ante* evaluation of the conformity with the legal framework behind it and the lack of enforcement rise the need to identify precise, enforceable parameters for facing the challenges posed by social robots arises. To this end, the role of the law can be crucial.

4.2 The binding and complete nature of the legal system

The main elements that distinguish the legal discipline from philosophy are: methodology, object of investigation and the limits they have to handle. In fact, legal argumentation cannot operate in a *vacuum*, for it takes place in a proper, self-referred system and lawyers are bound externally by fundamental principles, typically affirmed in Constitutions.

It could be argued that even not every legal norm is equally enforceable all over the world. Nevertheless, it is likewise true that, considering a given field of application, legal dispositions are binding all the people involved in it or, at least, those previously determined and indicated [65]. This confers homogeneity of solutions and treatments.

However, reasons of major complaints about the law as a tool for the regulation of new technologies are: (i) the long time needed for its formulation and concrete application, (ii) the difficult individuation of the proper time for an intervention, (iii) the rigidity of its statements. The processes of discussion, decision and entry into force of a new legislation are often considered inconceivable with the speed of scientific evolution [52]. The regulation could intervene too late, thus loosing its incisiveness and failing the aim to prevent the spread of potentially harmful devices. At the same time, even the choice regarding "when to regulate" has a decisive impact. In fact, acting too early could damage the very phase of experimentation, stopping a process which might need to be only corrected. Same observations could be done even with regards to already operative rules. If they are proved ineffective or inadequate, their modification or replacement is seen as laborious and not timely enough not to damage scientific research [35].

Such considerations are based on two basic misunderstandings. First of all, the fact that legal strict statements always undermine innovation, development and competitiveness. Actually, it is the uncertainty in regulation that, by definition, produces that very effect [4]. Furthermore, a relevant false belief is that emerging technologies – and, in particular, examples of embodied-AIs – would highlight legal gaps, so as to require the formulation of specific norms. On the contrary, the law constitutes a system that is complete *per se*, for it does not consist in the black-letter-law only. It regards norms, but even legal doctrine and judicial applications [63].

Therefore, legal interpretation can help find solutions to specific cases, without the need for specific rules [22]. This would be possible: (i) through the application of rules governing similar cases or similar matters (*analogia legis*), (ii) through the interpretation of the legislator's will, by means of the general princi-

ples of the legal system (*analogia iuris*), (iii) through elastic concepts, applicable in many different cases (general clauses, i.g. good faith).

This does not exempt from the possibility to question the adequacy of existing norms. Therefore, any chance to revise the actual legislation to correct undesirable, inefficient or even sub-efficient outcomes should be seized. It follows that the right question legal experts should try to answer is not if the law can play its part in the regulation of social robotics and AI, but how it can do that. In this view, the priority should be to guarantee both scientific progress and an efficient protection for human beings.

To this end, the attention could be focused on fundamental human rights, which are not proper of a specific ideology but, at the same time, have the advantage to be precise and not ambiguous, without the necessity of a narrow detailed definition [19].

5 Overcoming the dichotomous approach: the role of Human Dignity

So far we have described an articulate interweaving of plausible but not completely self-sufficient solutions. Therefore, it could be useful to introduce another perspective through which to address the theme of deception in social robotics and its possible implications: human dignity.

The doctrinal debate about the very nature of this concept is still open. More precisely, it is accused of vagueness, for it is often theorised as indemonstrable, imperative, inexpressible [21].

However - in spite of the lack of a specific definition - it has been considered as the "foundation of freedom, justice, and peace in the world" [3] by the United Nations General Assembly. In fact, human dignity is proper of each one of use, just because we all belong to the humankind and it constitutes the core of what it means to be "human" [16]. This is reflected in terms of rights and duties, but even much more so in terms of the perception individuals have of themselves, their environment, the others and of the way they can act and relate to this ecosystem. This aspect is crucial in addressing new technologies which, due to a daily bases interaction, can influence our intentions, awareness and the way we process, categorise and evaluate information, concepts, relations [7].

Its relevance and authority as a value is unquestionable. In fact, it plays a central role among fundamental rights, for it is the one which summarises all the others [88]. Moreover, its respect is not limited because of age, gender, religion, nationality, political convictions or any other subjective factor [60].

Concurrently, it is deeply rooted in the European tradition also for what lies outside the purely philosophical reflection, for it represents a legal concept. In fact, human dignity is considered the central principle of modern democracies [58]. It is recognised in many national norms and Constitutions around the world and it has become soon the very essence of both the European Court of Justice and of the European Court of Human Rights [58], as much as of EU treaties – especially the European Charter of Fundamental Human Rights – and of eminent

judgments in the Courts of Justice. With regard to this latter aspect, two of the most emblematic judicial cases are the German "*Peep-Show-Fall*" [18] and the French "*Jeux de nains*" [32]. In both of them, the Courts highlighted that every human being carries a fragment of the universal principle of dignity. As a consequence, diminishing one own value implies to reduce the one of all the affiliates. In such a view, human dignity may be used as an external limit to the exercise of other rights, including the right to self-determination. Historically, it has held this function other times, becoming the decisive instrument to prohibit - among others - slavery, torture, inhumane and degrading treatment, death penalty.

Therefore, it could represent an objective and external criterion, able to collect both the instances of philosophical speculation with regards to ethics and the non-dismissible and binding character of legal principles [33]. Thus, it would be functional in order to evaluate which types of technologies deserve to be favoured - for their correspondence to the reference values - and which ones to stem. This would be a starting point to guarantee that the technological development respects the centrality of human beings. In fact, human dignity can be used for: (i) testing the desirability of robotics applications *ex ante*, (ii) identifying – even in a case-by-case perspective – which principles should prevail, (iii) orienting innovation towards devices that allows to promote such values, (iv) allowing that they can be efficiently tested in the real world - not just in a laboratory - [44][45].

Furthermore, even admitting that this concept still have a certain level of abstraction, for what the regulation of social robotics is concerned this is not a negative aspect. It could have two functions: (i) guaranteeing flexibility and (ii) shaping mandatory norms. In fact, flexibility allows this principle to adapt more efficiently to the manifold variety of existing technologies and their unceasing innovation and development. At the same time, the already demonstrated effectiveness as a legal principle is essential for making it a rigorous, binding regulatory tool, not modulable on the base of the ethical framework taken into account [33].

6 Discussion and final remarks

This paper assessed the theme of deception in social robotics, underlining the need to identify an objective criterion to balance the demand for acceptability - to foster innovation - and the necessity to protect users' material and psychological integrity.

To such an end, the traditional juxtaposition between ethical and legal perspectives was presented, so as to underline their structural differences – mainly in terms of methodology and scope –. This is intended to clarify that none of them should be removed from the debate on the protection of new technologies' users. Nevertheless, they should play a different role in pursuing such a goal.

Ethics may be inspirational from a political or economic point of view. It can promote the introduction of a new or reformed legislation, the implementation of

companies' policies, the development of new awareness campaigns in the public [33]. Moreover, ethical principles are, in many cases, useful to overcome legal theories' limitations or to better understand and to convey the *ratio* on the base of legal reasoning [33]. Hence, they can live in a functional relationship with the law and help overcoming the strict boundaries of its formalities. Nonetheless, they should always be seen as an instrument to reach a goal, not the goal itself.

For its part, the law has the merit of being a complete, binding and enforceable system, deeply rooted in fundamental principles. Thus, it can be useful to inspire the design of socially competent devices and to evaluate their effects not only on the rights of the users, but also on those of all the members of society.

Given the peculiarities and variety of the theme here analysed, this paper suggests the possibility to rely on a third option: fundamental human rights. In particular, human dignity could be used as the external criterion through which to approach the regulation.

The reason why human dignity - among fundamental rights - should have such a role lies on the fact that it is recognised as the right upon which all the others are grounded [29]. Moreover, if on one hand all of them are, by definition, inalienable, universal and interdependent, on the other some of them can be limited. This is the case of exceptional and urgent circumstances, in order to safeguard a specific right - or group of right - which has to be considered temporarily prevalent. In addition, some of them can be subjected to forms of "inherent" limitations, once again closely linked to contingent circumstances. This is the case, among others, of the right to personal freedom that can be limited if an individual is detained in prison because convicted of a crime. Among the fundamental rights that have to be guaranteed at all times and no matter what, with regards to the theme of this discussion, human dignity is the one that should prevail. Furthermore, as it was briefly underline in Section 5, it has already been effectively used at a regulatory and jurisprudential level as a tool to limit competing rights, even the principle of self-determination.

Indeed, despite the claims of conceptual vagueness, it is legally binding, common to everyone because of their belonging to humankind, adaptable but not dismissible. Such a flexibility could be essential in order to face the challenges that deception in social robotics can – and will – pose. In fact, this theme is still controversial in the engineering and robotic field, as it emerges form the scenarios of interaction here presented. Nowadays the ability of a social robot to deceive – inducing the user to create an emotional and subconscious bond with the machine – is considered "central to AI as the circuits and software that make it run" [53]. However, (i) the qualification of such a dynamic and (ii) the scope for action to protect peopled involved are still reasons for debate. Actually, such technologies are already part of our reality, although they are not yet so widespread as to allow neither a sufficiently well-stocked collection of concrete cases, nor an in-depth investigation of their long-term effects. This makes difficult for philosophers and legal scholars to have an univocal perception of the issues raised from this kind of technology and to face them in an effective and appropriate way. For this very reason, it is crucial to promote an

integrated, multidisciplinary approach, able to take into consideration both the specificity of the social robotics field and the importance of a human-centered technological development. The aim of this discussion, in fact, is not to condemn the implementation of machines with social features *tout court*, but to suggest the need to draw a line between beneficial and risky contexts, as much as the one to investigate whether and to what extent to enhance robotic deceptive behavior towards the user.

This cannot be done in the logic of maximising the utility and minimising disutility only - like an utilitarian perspective seems to advocate -. In the design, development and use of a technological application, the respect for human beings and for the totality of their rights and values should be considered before the mere positive consequences of adopting such a device. Said otherwise, the price to be paid for a beneficial outcome cannot be the sacrifice of any human principle, value, right.

To this end, human dignity can be not just a parameter that can be harmed or a core element that need to be protected or guaranteed, but rather a concrete governance instrument.

In light of the above, further investigation of this principle is needed, to better define the issues that machines' manipulation and deception can cause in the various contexts in which they are used. Thus, it could be possible to define more concretely and precisely the impact assessment of human dignity in the regulation of new technologies.

7 Acknowledgements

The author acknowledges that this work has received funding from the Alma Mater Research Institute for Human-Centered AI, "Law, Science and Technology Joint Doctorate", University of Bologna.

References

1. Act, A.I.: Proposal for a regulation of the european parliament and the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts. EUR-Lex-52021PC0206 (2021)
2. AI, H.: High-level expert group on artificial intelligence (2019)
3. Assembly, U.G., et al.: Universal declaration of human rights. UN General Assembly **302**(2), 14–25 (1948)
4. Bahmani-Oskooee, M., Saha, S.: On the effects of policy uncertainty on stock prices: an asymmetric analysis. Quant Financ Econ **3**, 412–424 (2019)
5. Bertolini, A.: Robots as products: the case for a realistic analysis of robotic applications and liability rules. Law, innovation and technology **5**(2), 214–247 (2013)
6. Bertolini, A.: Human-robot interaction and deception. Osservatorio del diritto civile e commerciale **7**(2), 645–659 (2018)
7. Bisol, B., Carnevale, A., Lucivero, F.: Diritti umani, valori e nuove tecnologie. Metodo. International studies in phenomenology and philosophy **2**, 235–252 (2014)

8. Carli, R., Najjar, A.: Rethinking trust in social robotics. arXiv preprint arXiv:2109.06800 (2021)
9. Carotenuto, A.: Senso e contenuto della psicologia analitica. Bollati Boringhieri (1990)
10. Chugani, H.T., Behen, M.E., Muzik, O., Juhász, C., Nagy, F., Chugani, D.C.: Local brain functional activity following early deprivation: a study of postinstitutionalized romanian orphans. Neuroimage **14**(6), 1290–1301 (2001)
11. Coeckelbergh, M.: How to describe and evaluate "deception" phenomena: recasting the metaphysics, ethics, and politics of icts in terms of magic and performance and taking a relational and narrative turn. Ethics and Information Technology **20**(2), 71–85 (2018)
12. Columbus, L.: Mckinsey's state of machine learning and ai, 2017. Forbes. Available online: https://www. forbes. com/sites/louiscolumbus/2017/07/09/mckinseys-state-of-machine-learning-and-ai-2017 (accessed on 17 December 2020) (2017)
13. Danaher, J.: Robot betrayal: a guide to the ethics of robotic deception. Ethics and Information Technology **22**(2), 117–128 (2020)
14. Dautenhahn, K.: Socially intelligent robots: dimensions of human–robot interaction. Philosophical transactions of the royal society B: Biological sciences **362**(1480), 679–704 (2007)
15. Di Dio, C., Manzi, F., Peretti, G., Cangelosi, A., Harris, P.L., Massaro, D., Marchetti, A.: Shall i trust you? from child–robot interaction to trusting relationships. Frontiers in psychology **11**, 469 (2020)
16. Dicke, K.: The founding function of human dignity in the universal declaration of human rights. In: The concept of human dignity in human rights discourse, pp. 111–120. Brill Nijhoff (2001)
17. Diehl, J.J., Schmitt, L.M., Villano, M., Crowell, C.R.: The clinical use of robots for individuals with autism spectrum disorders: A critical review. Research in autism spectrum disorders **6**(1), 249–262 (2012)
18. Dreier, H.: Die, guten sitten "zwischen normativität und faktizität. In: Gedächtnisschrift für Theo Mayer-Maly, pp. 141–158. Springer (2011)
19. Dworkin, R.: Taking rights seriously harvard university press. Cambridge, Mass (1977)
20. van Est, Q., Gerritsen, J., Kool, L.: Human rights in the robot age: Challenges arising from the use of robotics, artificial intelligence, and virtual and augmented reality. Technology, Innovation & Society (2017)
21. Fabre-Magnan, M.: La dignité en droit: un axiome. Revue interdisciplinaire d'études juridiques **58**(1), 1–30 (2007)
22. FH Easterbrook, C., the Law of the Horse: The university of chicago legal forum (1996)
23. Floreano, D., Mitri, S., Magnenat, S., Keller, L.: Evolutionary conditions for the emergence of communication in robots. Current biology **17**(6), 514–519 (2007)
24. Floridi, L.: The fourth revolution: How the infosphere is reshaping human reality. OUP Oxford (2014)
25. Floridi, L.: Soft ethics and the governance of the digital. Philosophy & Technology **31**(1), 1–8 (2018)
26. Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., Luetge, C., Madelin, R., Pagallo, U., Rossi, F., et al.: Ai4people—an ethical framework for a good ai society: opportunities, risks, principles, and recommendations. Minds and Machines **28**(4), 689–707 (2018)
27. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots. Robotics and autonomous systems **42**(3-4), 143–166 (2003)

28. Fukuyama, F.: Our posthuman future: Consequences of the biotechnology revolution. Farrar, Straus and Giroux (2003)
29. Gilabert, P.: Human dignity and human rights. Oxford University Press, USA (2019)
30. de Graaf, M.M.A., Allouch, S.B.: The influence of prior expectations of a robot's lifelikeness on users' intentions to treat a zoomorphic robot as a companion. International Journal of Social Robotics **9**(1), 17–32 (2017)
31. Grodzinsky, F.S., Miller, K.W., Wolf, M.J.: Developing automated deceptions and the impact on trust. Philosophy & Technology **28**(1), 91–105 (2015)
32. Gros, M.: Il principio di precauzione dinnanzi al giudice amministrativo francese. Il principio di precauzione dinnanzi al giudice amministrativo francese pp. 709–758 (2013)
33. Harris, I., Jennings, R.C., Pullinger, D., Rogerson, S., Duquenoy, P.: Ethical assessment of new technologies: a meta-methodology. Journal of Information, Communication and Ethics in Society (2011)
34. Hegel, F., Krach, S., Kircher, T., Wrede, B., Sagerer, G.: Understanding social robots: A user study on anthropomorphism. In: RO-MAN 2008-The 17th IEEE International Symposium on Robot and Human Interactive Communication. pp. 574–579. IEEE (2008)
35. Holder, C., Khurana, V., Harrison, F., Jacobs, L.: Robotics and law: Key legal and regulatory implications of the robotics age (part i of ii). Computer law & security review **32**(3), 383–402 (2016)
36. Howard, A., Tapus, A., Kajitani, I.: Socially assistive robots [from the guest editors]. IEEE Robotics & Automation Magazine **26**(2), 10–110 (2019)
37. Huijnen, C., Lexis, M., De Witte, L.: Robots as new tools in therapy and education for children with autism. International Journal of Neurorehabilitation **4**(4), 1–4 (2017)
38. Isaac, A., Bridewell, W.: Why robots need to deceive (and how). Robot ethics **2**, 157–172 (2017)
39. Johnson, D.G., Verdicchio, M.: Why robots should not be treated like animals. Ethics and Information Technology **20**(4), 291–301 (2018)
40. Kaminski, M.E., Rueben, M., Smart, W.D., Grimm, C.M.: Averting robot eyes. Md. L. Rev. **76**, 983 (2016)
41. Kanda, T., Sato, R., Saiwaki, N., Ishiguro, H.: A two-month field trial in an elementary school for long-term human–robot interaction. IEEE Transactions on robotics **23**(5), 962–971 (2007)
42. Kass, L.R.: Ageless bodies, happy souls: biotechnology and the pursuit of perfection. The New Atlantis **1**, 9–28 (2003)
43. Kitwood, T.M.: Dementia reconsidered: The person comes first. Open university press (1997)
44. Koops, B.J.: Concerning 'humans' and 'human'rights. human enhancement from the perspective of fundamental rights. In: Engineering the Human, pp. 165–182. Springer (2013)
45. Kritikos, M.: Artificial intelligence ante portas: Legal & ethical reflections. European Parliamentary Research Service (2019)
46. Lee, K.M., Peng, W., Jin, S.A., Yan, C.: Can robots manifest personality?: An empirical test of personality recognition, social responses, and social presence in human–robot interaction. Journal of communication **56**(4), 754–772 (2006)
47. Machinery, C.: Computing machinery and intelligence-am turing. Mind **59**(236), 433 (1950)

48. Matarić, M.J.: Socially assistive robotics: Human augmentation versus automation. Science Robotics **2**(4), eaam5410 (2017)
49. Matthias, A.: Robot lies in health care: When is deception morally permissible? Kennedy Institute of Ethics Journal **25**(2), 169–162 (2015)
50. Metzinger, T.: Ethics washing made in europe. Der Tagesspiegel **8** (2019)
51. Mittelstadt, B.: Principles alone cannot guarantee ethical ai. Nature Machine Intelligence **1**(11), 501–507 (2019)
52. Moses, L.B.: Agents of change: How the law'copes' with technological change. Griffith Law Review **20**(4), 763–794 (2011)
53. Natale, S., et al.: Deceitful media: Artificial Intelligence and social life after the Turing Test. Oxford University Press, USA (2021)
54. Nelson, C.A., Zeanah, C.H., Fox, N.A., Marshall, P.J., Smyke, A.T., Guthrie, D.: Cognitive recovery in socially deprived young children: The bucharest early intervention project. Science **318**(5858), 1937–1940 (2007)
55. Nussbaum, M.C.: Frontiers of justice: Disability, nationality, species membership. Belknap Press Cambridge, MA (2006)
56. Orwell, G.: Nineteen eighty-four (1949). The complete novels **7** (1990)
57. Ozaeta, L., Graña, M., Dimitrova, M., Krastev, A.: Child oriented storytelling with nao robot in hospital environment: preliminary application results. Problems of Engineering Cybernetics and Robotics **69**, 21–29 (2018)
58. O'Mahony, C.: There is no such thing as a right to dignity. International Journal of Constitutional Law **10**(2), 551–574 (2012)
59. Palmerini, E., Bertolini, A., Battaglia, F., Koops, B.J., Carnevale, A., Salvini, P.: Robolaw: Towards a european framework for robotics regulation. Robotics and autonomous systems **86**, 78–85 (2016)
60. Resta, G.: La disponibilità dei diritti fondamentali ei limiti della dignità (note a margine della carta dei diritti). Riv. dir. civ **2**, 801–848 (2002)
61. Robinson, H., MacDonald, B., Kerse, N., Broadbent, E.: The psychosocial effects of a companion robot: a randomized controlled trial. Journal of the American Medical Directors Association **14**(9), 661–667 (2013)
62. Šabanović, S., Bennett, C.C., Chang, W.L., Huber, L.: Paro robot affects diverse interaction modalities in group sensory therapy for older adults with dementia. In: 2013 IEEE 13th international conference on rehabilitation robotics (ICORR). pp. 1–6. IEEE (2013)
63. Sacco, R.: Che cos' è il diritto comparato, vol. 2. Giuffrè (1992)
64. Salem, M., Eyssel, F., Rohlfing, K., Kopp, S., Joublin, F.: To err is human (-like): Effects of robot gesture on perceived anthropomorphism and likability. International Journal of Social Robotics **5**(3), 313–323 (2013)
65. Sartor, G.: Artificial intelligence and human rights: Between law and ethics. Maastricht Journal of European and Comparative Law **27**(6), 705–719 (2020)
66. Savela, N., Turja, T., Oksanen, A.: Social acceptance of robots in different occupational fields: A systematic literature review. International Journal of Social Robotics **10**(4), 493–502 (2018)
67. Scassellati, B., Admoni, H., Matarić, M.: Robots for use in autism research. Annual review of biomedical engineering **14**, 275–294 (2012)
68. Sharkey, A., Sharkey, N.: Children, the elderly, and interactive robots. IEEE Robotics & Automation Magazine **18**(1), 32–38 (2011)
69. Sharkey, A., Sharkey, N.: Granny and the robots: ethical issues in robot care for the elderly. Ethics and information technology **14**(1), 27–40 (2012)
70. Sharkey, A., Sharkey, N.: We need to talk about deception in social robotics! Ethics and Information Technology pp. 1–8 (2020)

71. Shim, J., Arkin, R.C.: A taxonomy of robot deception and its benefits in hri. In: 2013 IEEE international conference on systems, man, and cybernetics. pp. 2328–2335. IEEE (2013)
72. Shim, J., Arkin, R.C.: Other-oriented robot deception: How can a robot's deceptive feedback help humans in hri? In: International Conference on Social Robotics. pp. 222–232. Springer (2016)
73. Short, E., Hart, J., Vu, M., Scassellati, B.: No fair!! an interaction with a cheating robot. In: 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI). pp. 219–226. IEEE (2010)
74. Shukla, J., Cristiano, J., Amela, D., Anguera, L., Vergés-Llahí, J., Puig, D.: A case study of robot interaction among individuals with profound and multiple learning disabilities. In: International Conference on Social Robotics. pp. 613–622. Springer (2015)
75. Skorupinski, B., Ott, K.: Technology assessment and ethics. Poiesis & Praxis **1**(2), 95–122 (2002)
76. Sorell, T., Draper, H.: Second thoughts about privacy, safety and deception. Connection Science **29**(3), 217–222 (2017)
77. Sparrow, R.: The march of the robot dogs. Ethics and information Technology **4**(4), 305–318 (2002)
78. Stile, G.C.: Transumanesimo. un'introduzione all'idea di evoluzione autodiretta. Laboratorio dell'ISPF ISSN 1824-9817 (2015)
79. Stradella, E.: La regolazione della robotica e dell'intelligenza artificiale: il dibattito, le proposte, le prospettive. alcuni spunti di riflessione. Media Laws **1**, 1–20 (2019)
80. Thompson, E., Zahavi, D.: Price, dd, and barrell, jj (2012). develop. Susan Blackmore and p. 489 (2012)
81. Turkle, S.: Alone together: Why we expect more from technology and less from each other. Hachette UK (2017)
82. Turkle, S.: Why these friendly robots can't be good friends to our kids. Washington Post, December (2017)
83. Vandemeulebroucke, T., de Casterlé, B.D., Gastmans, C.: The use of care robots in aged care: A systematic review of argument-based ethics literature. Archives of gerontology and geriatrics **74**, 15–25 (2018)
84. Wagner, A.R., Arkin, R.C.: Acting deceptively: Providing robots with the capacity for deception. International Journal of Social Robotics **3**(1), 5–26 (2011)
85. Wagner, B.: Ethics as an escape from regulation. from "ethics-washing" to ethics-shopping? In: Being Profiled, pp. 84–89. Amsterdam University Press (2018)
86. Wallach, W., Allen, C.: Moral machines: Teaching robots right from wrong. Oxford University Press (2008)
87. Wilson, D., Sperber, D.: On grice's theory of conversation. Conversation and discourse pp. 155–78 (1981)
88. Zatti, P.: Note sulla semantica della dignità. Maschere del diritto volti della vita pp. 24–49 (2009)
89. Złotowski, J., Proudfoot, D., Yogeeswaran, K., Bartneck, C.: Anthropomorphism: opportunities and challenges in human–robot interaction. International journal of social robotics **7**(3), 347–360 (2015)

Transfer Learning and Curriculum Learning in Sokoban

Zhao Yang[1], Mike Preuss[2], and Aske Plaat[3]

[1] LIACS, Leiden University, the Netherlands
z.yang@liacs.leidenuniv.nl
[2] LIACS, Leiden University, the Netherlands
m.preuss@liacs.leidenuniv.nl
[3] LIACS, Leiden University, the Netherlands
aske.plaat@gmail.com

Abstract. Transfer learning can speed up training in machine learning, and is regularly used in classification tasks. It reuses prior knowledge from other tasks to pre-train networks for new tasks. In reinforcement learning, learning actions for a behavior policy that can be applied to new environments is still a challenge, especially for tasks that involve much planning. Sokoban is a challenging puzzle game. It has been used widely as a benchmark in planning-based reinforcement learning. In this paper, we show how prior knowledge improves learning in Sokoban tasks. We find that reusing feature representations learned previously can accelerate learning new, more complex, instances. In effect, we show how curriculum learning, from simple to complex tasks, works in Sokoban. Furthermore, feature representations learned in simpler instances are more general, and thus lead to positive transfers towards more complex tasks, but not vice versa. We have also studied which part of the knowledge is most important for transfer to succeed, and identify which layers should be used for pre-training.[4]

Keywords: Reinforcement learning · Transfer learning · Sokoban.

1 Introduction

Humans are good at reusing prior knowledge when facing new problems. As a consequence, we learn new tasks quickly, a skill of great interest in machine learning. In the human brain, information received by our sensors is first transformed into different forms, and different types of transformed information are stored in different areas of our brain. When another problem arrives later on, we retrieve useful information and adjust it to better suit solving this new problem. The knowledge stored in artifical neural networks is also re-usable and transferable [31]. In supervised learning, pre-trained networks are commonly applied

[4] Codes we used for this work can be found at https://github.com/yangzhao-666/TLCLS

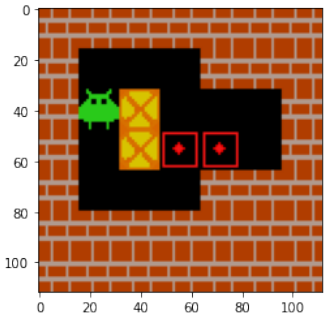


Fig. 1: An example instance of Sokoban.

in computer vision [17,25] and natural language processing [3,9]. Feature representations learned from images or words overlap to some extent, which makes such feature representations reusable and transferable. In reinforcement learning (RL), transfer learning is relatively new, although with the spread of deep neural networks, reusing pre-trained models becomes possible in RL as well [1,7]. Transfer learning works well in RL for recognition tasks, but tasks that rely heavily on planning are harder.

In this paper, we study transfer learning of behavior in Sokoban, a popular RL game in which planning is important [10,12]. It has already been proved that Sokoban is PSPACE-complete [8] and NP-hard problem [10]. An example instance from [22] is shown in Fig. 1. The goal of Sokoban is to control a warehouse worker that pushes all boxes onto targets. Sokoban is a challenging game where one wrong move can lead to a dead end (after a box has been pushed, it can not be pulled, and we cannot undo an inadvertent push). This non-reversibility is known to make games harder for AI agents [5]. Learning to solve Sokoban tasks is a challenge, especially in the multi-box scenario. For humans, if we have learned the basics of Sokoban (what is a box, what can an agent do), and if we are faced with a new, more complex instance, then we immediately focus on the new challenges in the instance, rather than re-learning the basics again. This building on prior knowledge saves time in the problem-solving process.

We investigate if we can achieve this kind of pretraining/fine-tuning learning in an RL agent. Our main hypothesis is that feature representations learned in Sokoban instances can be reused to improve solving other instances, and that features learned in simpler instances are more general and better transferable. We test this hypothesis by means of different experiments, in which parts of the neural network that has previously been trained on one type of instances (e.g. one box one target) are taken over (unchanged) to a new type of instances (e.g. two boxes two targets), whereas the remaining part of the network is trained on these new instances from scratch. The overall idea is that we see successful transfer if the preserved knowledge (in terms of network layers) leads to a faster learning process on the new problem type.

The main contributions of this paper are as follows: First, we show that feature representations learned in simple Sokoban instances can accelerate learning in more complex instances, indicating that curriculum learning can be used in Sokoban. Second, feature representations of simpler instances are more general and reusable than features learned in more complex instances. Third, our results confirm that in RL lower layers learn more general features. Interestingly, in some cases the best performance is achieved when more specific features are transfered, when source task and target task are similar enough to support these more specific features. Fourth, we found negative transfer from a simple supervised learning task, which tells us that choice and design of the source tasks are crucial. Fifth, we show that transferring top-fully-connected layers will not only be unhelpful but also harmful to the learning. We also used popular visualization techniques to explore potential reasons for successful transfers, which we explain in detail. Our code and test environments will be made available after blind review.

The paper is structured as follows: we first briefly review related work on transfer learning and Sokoban in the next section; then the environment and methods we are using are described in Section 3; Section 4 shows the experimental settings and results; in the last section, we conclude our work and discuss some potential future directions.

2 Related Work

De la Cruz et al. [6] studied the reuse of feature representations between two similar games: Breakout and Pong, using Deep Q Network(DQN). They used a 3-layer convolutional network. Weights learned in one game were transferred to improve learning the other game; results showed positive transfer of features between the different games. Pong and Breakout do not require planning; in our experiments, in Sokoban, we study how a curriculum of simpler instances can benefit the learning of complex instances. Spector et al. [26] used self-transfer in a DQN grid-world task to identify which parts should be transferred and which parts should be fixed, showing significant benefit of knowledge transfer.

Sokoban is a planning task that has been used as a benchmark for model-based reinforcement learning [22,16]. It It has also been used in model-free RL [14,15], achieving performance competitive with model-based methods. The efficiency of AlphaZero-style curriculum learning has been shown by solving hard single Sokoban instances [11,12]. Previous works were aimed at solving single Sokoban instances; our paper focuses on the transferability of learned knowledge among *different* instances.

This transferability of learned feature representations was first studied in image classification problems [31]. It was shown that bottom layers in Convolutional Neural Networks(CNNs) extract more general features while ones extracted from back layers are more specific. In this paper, we verify this idea under RL settings.

Reinforcement learning [27,21] aims to reinforce behaviors of the learning agent by rewarding signals obtained from interactions with the environment. It

has reached super-human performance in games such as Go [24], StarCraft [29,20], as well as Atari games [2] and robotic tasks. In this paper we follow the conventional MDP notation for RL [27].

Transfer learning reuses prior knowledge to improve the learning efficiency or performance in new tasks [30,28]. In reinforcement learning, higher-level knowledge such as macro actions, skills and lower-level knowledge such as reward functions, policies could be transferred. Transferring learned knowledge could take different approaches, such as reward shaping [4], learning from demonstration [19] and policy reuse [13].

3 Experimental Setup

The environment used in the paper is the Gym environment for Sokoban [23]; for the agent algorithms we follow Weber et al. [22]. Examples are shown in Fig. 2. The game is solved by controlling the agent (green sprite) to push all boxes (yellow squares) onto corresponding targets(red squares). There's no hint about which boxes should on which targets, and boxes can only be pushed; some actions are irreversible, and can leave the game in an unsolvable state. The difficulty of the game can be increased easily by putting more boxes as well as targets into generated rooms. The agent can go up, down, left, and right. The agent gets a final reward of 10 by pushing all boxes on targets. Pushing a box on a target will result a reward of 1 and a penalty of -1 for pushing a box off a target. We also give a small penalty of 0.1 for each step the agent takes.

We perform three types of experiments: (1) related tasks (source and target tasks are both RL tasks, while source tasks are to solve n-boxes Sokoban instances and target tasks are to solve m-boxes Sokoban instances, where $n \neq m$), (2) different tasks (source tasks are supervised learning(SL) tasks and target tasks are reinforcement learning(RL) tasks), and (3) different texture appearance(source and target tasks are both RL tasks, while source tasks are to solve original Sokoban instances and target tasks are to solve Sokoban instances with different texture appearance). The agent was first pre-trained on source tasks and then fine-tuned on target tasks. RL tasks are to solve 100 randomly generated n-boxes Sokoban instances. SL tasks are to recognize the location of the agent in Sokoban instances.

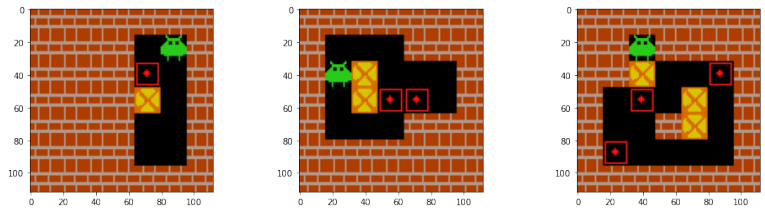


Fig. 2: Examples of Sokoban instances, increasing in difficulty from 1 box and 1 target to 3 boxes and 3 targets

The overall statistics of the maps are shown in Fig. 3. As the number of objectives increases, the number of steps for the optimal solution also increases, and so does the difficulty of solving the game.

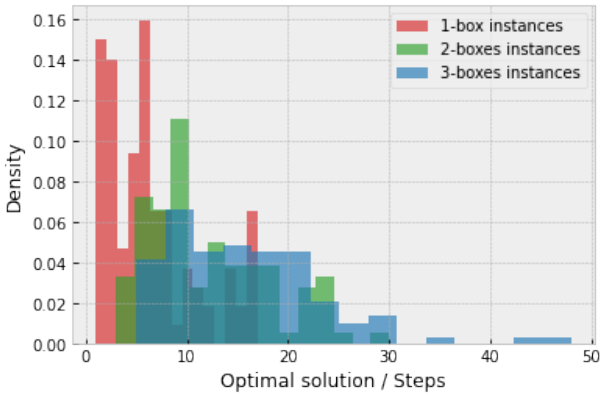


Fig. 3: Distribution of optimal solutions in different Sokoban instances.

3.1 Neural Network Architecture

The neural network we employ is taken from the DeepMind baseline [22] directly without hyper-parameter tuning. The model consists of 3 convolutional (Conv) layers with kernel size 8×8, 4×4, 3×3, strides of 4, 2, 1, and number of output channels 32, 64, 64. This is followed by a fully connected (FC) hidden layer with 512 units. The outputs of this FC layer will be fed into two heads: one for outputting the policy logits and one for outputting the state value. This is one of the most commonly-used architectures in RL, we selected it also in order to show what can be achieved with popular architecture. Details of architecture and hyper parameters we employ are found in Table 1.

Table 1: Hyper-parameters of the neural network and training.

learning rate	$7 \cdot 10^{-4}$
discount factor	0.99
entropy coefficient	0.1
value loss coefficient	0.5
eps in RMSprop	10^{-5}
alpha in RMSprop	0.99
rollout storage size	5
No. of environments for collecting trajectories	30

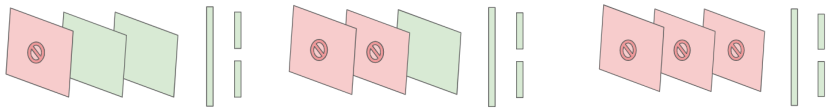


Fig. 4: Three different transfer approaches, red layers are fixed while green layers are trainable. They correspond $k = 1, 2, 3$ from left to right respectively.

3.2 Transfer Approach

The main idea of our transfer approach is to reuse feature representations from source tasks learned by the Conv layers in new unseen target tasks. As detailed in the last sub-section, our model consists of 3 Conv layers and 2 FC layers. The feature representations were transferred to new tasks by copying the weights of the first k Conv layers trained in source tasks (where there are n_s boxes/targets) to initialize the new learning model in target tasks (where there are n_t boxes/targets). Then we froze these weights (they were no longer trainable) and retrained the remaining part of the model. In our experiments, $k \in \{1, 2, 3\}$, $n_s \in \{1, 2, 3\}$, $n_t \in \{1, 2, 3\}$. Please refer to Fig. 4 for an explanation of this approach. Different squares represent different layers of our neural network. The first 3 layers are Conv layers and the last two are FC layers. Reds are weights taken from pre-trained model and fixed, greens are weights reinitialized and trainable.

Solved ratios were used for evaluating agents, and evaluation executes every 1,000 environment steps. 20 randomly selected test instances were performed by the current learning agent. We say the transfer is *positive* when the performance with the transfer is better than without (training from scratch), and *negative* when the performance with the transfer is worse than without.

4 Experiments

We designed experiments with different source, target tasks and k, in order to verify the hypotheses we proposed. We experimented with Sokoban instances with 1, 2, and 3 boxes. All experiments were run for 1 million environment steps. We use abbreviations for each experiment. For instance, **s1t1k1** means **s**ource tasks are **1**-box instances, **t**arget tasks are **1**-box instances and we transfer and fix the **1**(first) layer. Exceptions are **sPt1k1** and **s1t1fc_game2**. **sPt1k1** stands for the source task is a supervised learning prediction task, and target task is the RL task over 1-box instances while we only keep the first layer. **s1t1fc_game2** is that the source and target tasks are both RL tasks over 1-box instances, but we transfer fully connected layers to instances with different appearance. The neural networks were trained using Advantage Actor Critic(A2C), a single threaded variant of A3C [18]. All experiments were performed 5 times with different random seeds, and figures were drawn using averaged results with 0.95

confidence interval. Heavy fluctuations were caused by irreversible actions, one irreversible action during the game could make the whole game unsolvable.

4.1 Transfer Among Related Tasks

Related tasks are tasks where the only difference between source and task is the difficulties of instances, i.e. the number of boxes and targets. (Recall that both source and task are trained on 100 different map-layouts, in all experiments.)

Fig. 5 and Fig. 6 show results for training on 1-box, 2-boxes, 3-boxes instances with reusing features learned in different tasks, and we fix $k = 3$. All results showed that transferring feature representations learned in single-box instances is positive. Performance of agents (s1t1k3, s1t2k3, s1t3k3) who are using features learned from single-box instances always outperform other agents, including agents training from scratch and using features learned from other instances. The transfer, however, is not 'bi-directional', feature representations learned in multiple-box instances could not be successfully transferred to the learning in single-box instances. Their performance (s2t1k3, s3t1k3) converged to a relatively low solved ratio, which indicates that transferred features are not suitable for single-box instances. Just as humans learn more general knowledge in simpler cases, our agents also showed that the knowledge learned from single-box instances is more general and transferable than ones learned in multiple-box instances.

To further enhance performances of transferring features learned in single-box instances, we tried different k. We expected that the performance will be the best when $k = 1$ since the first layer learn the most general features. However, the results in Fig. 7 instead show that not $k = 1$ but $k = 2$ (s1t2k2, s1t3k2)

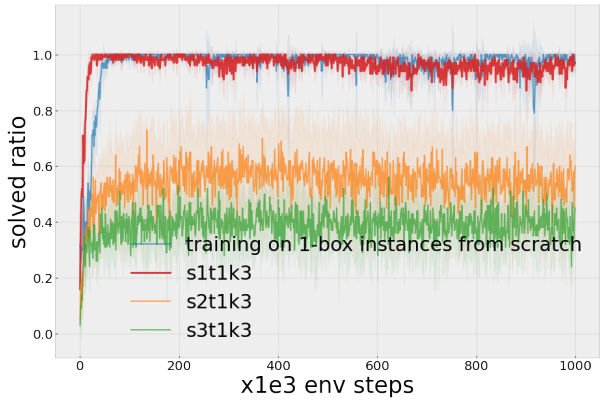


Fig. 5: Performance of transferring feature representations learned in 1-box, 2-boxes, 3-boxes instances to learning in 1-box with $k = 3$. $n_s = 1, 2, 3$, $n_t = 1$, $k = 3$. Pre-training on 1-box instances is much better than pre-training on 2 or 3 box instances when training new 1-box instances.

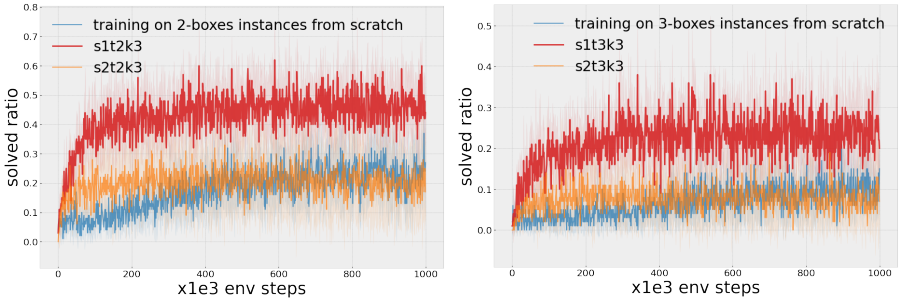


Fig. 6: Performance of transferring feature representations learned in 1-box, 2-boxes, 3-boxes instances to learning in 2-boxes (left) and 3-boxes (right) with $k = 3$. $n_s = 1, 2, 3$, $n_t = 2, 3$, $k = 3$.

perform the best. Similar to [6], features learned in the first 2 layers are still general enough for transfer; in addition, the difference between source tasks and target tasks is not as large as expected, and features learned between different instances are more overlapping than expected.

It is also interesting to see the influence of how many layers are fixed on the success of the transfer. In particular, we want to know whether a smaller k could change the negative transfer from multiple-box instances to single-box instances into positive. (We believe features from multiple and single-box instances are overlapping to some extent.) Results are shown in Fig. 8. We see that indeed the first layer (s2t1k1, s3t1k1) did learn enough general features from multiple-boxes instances to solve the single-box instances. Although agents with features only learned by the first layer could converge to decent performance in the end, the transfer is still negative. An interesting point is that $k = 3$ (s2t1k3) performs better than $k = 2$ (s2t1k2) when source tasks are 2-boxes instances. Note that $k = 2$ (s3t1k2) performs better than $k = 3$ (s3t1k3) when source tasks are 2-boxes instances. There are more overlapping features between the 2-boxes instances and single instances.

4.2 Transfer Among Different Tasks (SL/RL)

Feature representations learned from previous tasks can either be helpful or harmful. In the previous subsection we saw some positive transfer to related Sokoban tasks, in this subsection we study if transfer between supervised and reinforcement learning tasks works. We follow prior work, Anderson et al. [1] showed that features can be transfered from hand-crafted supervised learning(SL) tasks to reinforcement learning(RL). Their model was first trained to predict state dynamics of the environment, and then pre-trained hidden layers were helpful to accelerate solving RL tasks.

For transfer to different (randomly chosen) instances in Sokoban, we also formed a supervised task, which was to train a prediction model to recognize

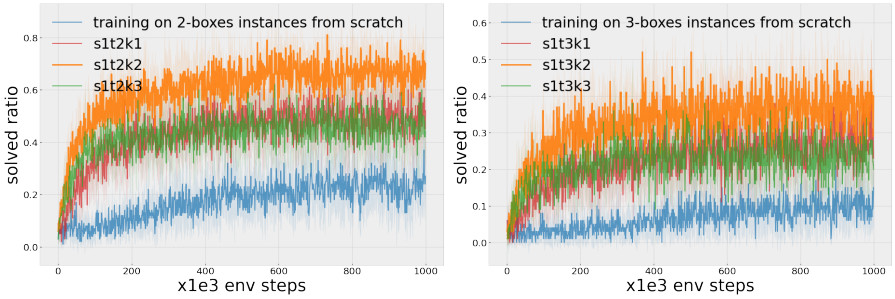


Fig. 7: Performance of transferring feature representations learned in 1-box instances to learning in 2-boxes (left) and 3-boxes (right) with different k. $n_s = 1$, $n_t = 2, 3$, $k = 1, 2, 3$.

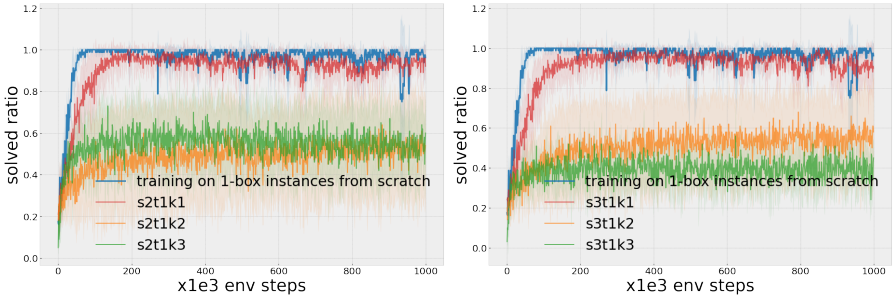
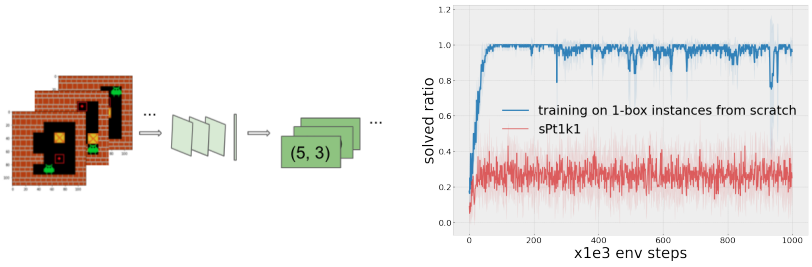


Fig. 8: Performance of transferring feature representations learned in 2-boxes (left) and 3-boxes (right) instances to learning in 1-box instances with different k. $n_s = 2, 3$, $n_t = 1$, $k = 1, 2, 3$.

the location of the agent, shown in Fig. 9a. When humans are solving Sokoban, we first need to know where the agent is before we draw up a plan. If we already know the location of objectives, the solving process could be faster. After the prediction model could correctly recognize where the agent is, we took feature representations of the trained model and plug them into a new agent. The first layer of learned features is fixed, and we only train the remaining part. Fig. 9b shows the performance of transferring and training from scratch. We find negative transfer for (sPt1k1): the performance is much worse compare with training from scratch.

4.3 Transfer To Different Appearance

Experiments we described in previous subsections were all trying to transfer Conv layers which learned feature representations. In the next experiment, we try to make the agent utilize another part of the learned model, which are back FC layers of the whole model. The source and target tasks were both single-



(a) (b)

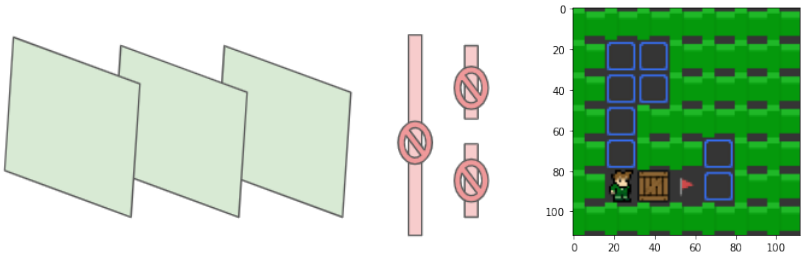
Fig. 9: (a): How SL tasks work. Input states and neural network will learn to predict locations of the agent. (b): Performance of training from scratch and training with transferred feature representations from SL tasks.

box instances, but the target tasks were instances with different appearances. Fig. 10b is an example. The maps used for two groups of tasks were the same, the only difference was how they look like, the appearance was changed, with different textures, and we call it Game2. Fig. 10a shows the transfer approach. We took FC layers trained in source tasks and fixed them, and retrained the remaining Conv layers. Since maps were the same, solutions of the instances were the same. When Conv layers learn new feature representations successfully, instances are solved then.

Fig. 11a shows the performance. One would expect that transferred FC layers(s1t1fc_game2) are faster because the agent only needs to learn new feature representations. However, the experiments did not show this result. Apparently, when the whole model is trained jointly, it has more flexibility to be trained into the final shape; when the last part of the model is fixed, the learning of the first part will be trying to cater for the last part in order to solve the problem, which made the learning slower.

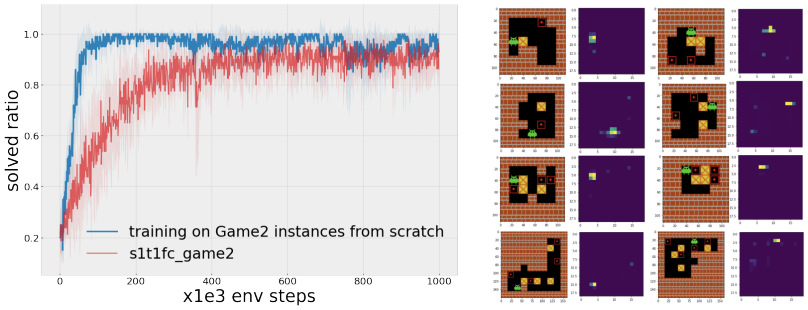
4.4 Visualizing Agent Detection

In order to better understand what the network learned, we provide a visualization. We follow Yosinski et al. who showed that convolutional neural networks can detect latent objectives without explicit labels [31]. We visualized a feature map of a trained neural network on 1-box RL tasks. Fig. 11b shows the latent 'agent detector' for Sokoban. The neural network automatically learned to detect the agent without giving any labels or information. Left rows are pixel inputs, right rows are outputs of one specific feature map. Yellow-green units are detected agents. We note that although the network was trained in single-box instances, it still performed quite well in multiple-box instances, which is a potential reason for the successful transfer. The agent's abilities that were learned in source tasks are useful in target tasks.



(a) (b)

Fig. 10: (a): Transfer approach for transfer to Game2. FC layers are taken from previously training and fixed, only conv layers will be retrained. (b): An example instance in Game2. We changed appearances in Game2 with different textures of objectives.



(a) (b)

Fig. 11: (a): Training on Game2 using transferred FC layers. Its performance is worse than training from scratch. (b): The agent detector. Outputs of the twenty third feature map of the first convolutional layer, which is an agent detector learned from 1-box instances, and it's still usable in multiple-boxes scenarios.

5 Conclusion and Future Work

Our experiments showed that in a reinforcement learning setting the agent in Sokoban can learn four characteristics that are similar to humans. (1) Feature representations learned previously can accelerate the new learning in other Sokoban instances. Knowledge learned in previous related tasks could be reused to accelerate new learning, transfer learning is occurring, creating an implicit learning curriculum. (2) Feature representations learned in single-box instances are more general, and are more effective for learning in multiple-boxes instances, but not vice versa. Knowledge learned in simpler tasks is more general and more effective, even in more complex tasks. Further experiments showed negative learning, that confirms these results. (3) Feature representations learned

in unrelated supervised learning tasks can hurt fine-tuning performance. If the learned knowledge is required to be helpful in new coming tasks, it's better to learn from similar tasks, otherwise the choice of tasks needs to be careful. (4) Fixing the top-fully-connected layers and retraining the bottom convolutional layers slows down learning and hurts performance. We conclude that learning should have explicit order, less flexibility will not only be unhelpful but also hurt the learning process and the performance.

Our experiments showed that with a simple 5-layer convolutions/fully connected network (based on DeepMind's baseline [22]), transfer learning and curriculum learning of behavior to occur in Sokoban. This is surprising, since Sokoban is a planning-heavy problem, for which one would expect more elaborate network architectures to be necessary. Reusing pre-trained feature representations in RL fields is not well studied, and to the best of our knowledge, these are the first results show transfer learning and curriculum learning with such a simple network in such a planning-heavy behavioral task. In the future, we would like to see more utilization of pre-trained feature representations and of the enire pre-trained model in RL. We believe that reusing pre-trained model can significantly improve data-efficient reinforcement learning.

Acknowledgement

The financial support to Zhao Yang is from the China Scholarship Council(CSC). Computation support is from ALICE and DSLab. The authors thank Hui Wang, Matthias Müller-Brockhausen, Michiel van der Meer, Thomas Moerland and all members from the Leiden Reinforcement Learning Group for helpful discussions.

References

1. Anderson, C.W., Lee, M., Elliott, D.L.: Faster reinforcement learning after pretraining deep networks to predict state dynamics. In: 2015 International Joint Conference on Neural Networks (IJCNN). pp. 1–7. IEEE (2015)
2. Badia, A.P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z.D., Blundell, C.: Agent57: Outperforming the atari human benchmark. In: International Conference on Machine Learning. pp. 507–517. PMLR (2020)
3. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
4. Brys, T., Harutyunyan, A., Taylor, M.E., Nowé, A.: Policy transfer using reward shaping. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. pp. 181–188 (2015)
5. Cook, M., Raad, A.: Hyperstate space graphs for automated game analysis. In: IEEE Conference on Games, CoG 2019, London, United Kingdom, August 20-23, 2019. pp. 1–8. IEEE (2019). https://doi.org/10.1109/CIG.2019.8848026, `https://doi.org/10.1109/CIG.2019.8848026`
6. De la Cruz, G., Du, Y., Irwin, J., Taylor, M.: Initial progress in transfer for deep reinforcement learning algorithms. In: 25th International Joint Conference on Artificial Intelligence (IJCAI). vol. 7 (2016)

7. Cruz Jr, G.V., Du, Y., Taylor, M.E.: Pre-training neural networks with human demonstrations for deep reinforcement learning. arXiv preprint arXiv:1709.04083 (2017)
8. Culberson, J.: Sokoban is pspace-complete (1997)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Dor, D., Zwick, U.: Sokoban and other motion planning problems. Computational Geometry **13**(4), 215–228 (1999)
11. Feng, D., Gomes, C.P., Selman, B.: A novel automated curriculum strategy to solve hard sokoban planning instances. Advances in Neural Information Processing Systems **33**, 3141–3152 (2020)
12. Feng, D., Gomes, C.P., Selman, B.: Solving hard AI planning instances using curriculum-driven deep reinforcement learning. CoRR **abs/2006.02689** (2020), `https://arxiv.org/abs/2006.02689`
13. Fernández, F., García, J., Veloso, M.: Probabilistic policy reuse for inter-task transfer learning. Robotics and Autonomous Systems **58**(7), 866–871 (2010)
14. Guez, A., Mirza, M., Gregor, K., Kabra, R., Racanière, S., Weber, T., Raposo, D., Santoro, A., Orseau, L., Eccles, T., et al.: An investigation of model-free planning. In: International Conference on Machine Learning. pp. 2464–2473. PMLR (2019)
15. Guez, A., Weber, T., Antonoglou, I., Simonyan, K., Vinyals, O., Wierstra, D., Munos, R., Silver, D.: Learning to search with mctsnets. In: International Conference on Machine Learning. pp. 1822–1831. PMLR (2018)
16. Hamrick, J.B., Friesen, A.L., Behbahani, F., Guez, A., Viola, F., Witherspoon, S., Anthony, T., Buesing, L.H., Velickovic, P., Weber, T.: On the role of planning in model-based deep reinforcement learning. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021 (2021)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
18. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. pp. 1928–1937. PMLR (2016)
19. Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Overcoming exploration in reinforcement learning with demonstrations. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 6292–6299. IEEE (2018)
20. Ontanón, S., Synnaeve, G., Uriarte, A., Richoux, F., Churchill, D., Preuss, M.: A survey of real-time strategy game AI research and competition in StarCraft. IEEE Transactions on Computational Intelligence and AI in Games **5**(4), 293–311 (2013)
21. Plaat, A.: Learning to Play: Reinforcement Learning and Games. Springer Verlag, Heidelberg, See `https://learningtoplay.net` (2020)
22. Racanière, S., Weber, T., Reichert, D.P., Buesing, L., Guez, A., Rezende, D., Badia, A.P., Vinyals, O., Heess, N., Li, Y., et al.: Imagination-augmented agents for deep reinforcement learning. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 5694–5705 (2017)
23. Schrader, M.P.B.: gym-sokoban. `https://github.com/mpSchrader/gym-sokoban` (2018)

24. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. nature **550**(7676), 354–359 (2017)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Spector, B., Belongie, S.: Sample-efficient reinforcement learning through transfer and architectural priors. arXiv preprint arXiv:1801.02268 (2018)
27. Sutton, R.S., Barto, A.G.: Reinforcement learning, An Introduction, Second Edition. MIT Press (2018)
28. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research **10**(Jul), 1633–1685 (2009)
29. Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature **575**(7782), 350–354 (2019)
30. Xu, W., He, J., Shu, Y.: Transfer learning and deep domain adaptation. In: Advances in Deep Learning. IntechOpen (2020)
31. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? Advances in Neural Information Processing Systems, pages 3320–3328 (2014)

Potential-based Reward Shaping in Sokoban

Zhao Yang[1], Mike Preuss[2], and Aske Plaat[3]

[1] LIACS, Leiden University, the Netherlands
z.yang@liacs.leidenuniv.nl
[2] LIACS, Leiden University, the Netherlands
m.preuss@liacs.leidenuniv.nl
[3] LIACS, Leiden University, the Netherlands
aske.plaat@gmail.com

Abstract. Learning to solve sparse-reward reinforcement learning problems is difficult, due to the lack of guidance towards the goal. But in some problems, prior knowledge can be used to augment the learning process. Reward shaping is a way to incorporate prior knowledge into the original reward function in order to speed up the learning. While previous work has investigated the use of expert knowledge to generate potential functions, in this work, we study whether we can use a search algorithm(A*) to automatically generate a potential function for reward shaping in Sokoban, a well-known planning task. The results showed that learning with shaped reward function is faster than learning from scratch. Our results also indicate that distance functions could be a suitable potential function for Sokoban. This work demonstrates the possibility of solving multiple instances with the help of reward shaping and results can be compressed into a single policy, which can be seen as the first phase towards training a general policy that is able to solve unseen instances.[4]

Keywords: Reinforcement Learning · Potential-based Reward Shaping · Sokoban.

1 Introduction

Sokoban is a well-known puzzle game that is often used as a benchmark for evaluating reinforcement learning (RL) agents [8,10]. It is a sparse reward task and also suffers from dead-ends: one bad action can render the whole instance unsolvable. Sokoban is deceptively simple, RL agents struggle to learn a behavior policy, unless they used planning as part of their learning effort. A simple example from [17] is shown in Fig. 1. Although this example has only three boxes, it might already be challenging to solve for humans.

Human problem solving used heuristics, rules of thumb that are based on experience, that work most of the time, but not always. Heuristics usually increase our ability to solve problems greatly. Reward shaping [5,13] is proposed

[4] Codes we used for this work can be found at
https://github.com/yangzhao-666/PbRSS

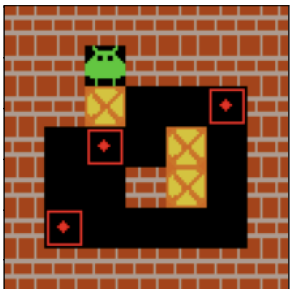


Fig. 1: An example of 3-boxes Sokoban instance.

for incorporating prior (heuristic) knowledge to accelerate learning in RL. It reshapes the original reward function by adding another reward function which is formed by prior knowledge in order to get an easy-learned reward function, that is often also more dense. Examples of prior knowledge are heuristics from context structures, demonstrations from experts, etc.

In this paper, we show that reward shaping can be applied to sparse reward tasks for faster learning. More accurately, we choose the distance function (automatically provided by the A* search algorithm) as the potential function, and subsequently performed potential-based reward shaping in Sokoban. Our results demonstrate that learning with shaped reward functions outperforms learning from scratch by a large margin. Neural networks are able to generalize to unseen tasks but require much training data, our reward shaping can be seen as the first step towards the final goal that aims to train an agent which is able to solve multiple unseen new Sokoban instances. With reward shaping, the ability of solving multiple instances is compressed into a single behavior policy without extensive training.

The paper is structured as follows: first we briefly describe related work in section 2; then details about our method are provided in section 3; followed by experimental design and results; lastly, we discuss limitations and potential future works of this paper, then draw conclusions in section 5.

2 Related Work

Reinforcement learning (RL) algorithms are used to solve decision making problems which could be formed into Markov Decision Process (MDPs), and they train policies by interacting with environments [14,19]. Recently, RL achieves super human performance in the board game Go [18], Atari games [1] and StarCraft [20]. In this section, we will briefly describe related work about both potential-based reward shaping and Sokoban.

2.1 Reward Shaping

Reward shaping offers a way to add useful information to the reward function of the original MDP. By reshaping, the original sparse reward function will be denser and is more easily-learned. The heuristics can come from different sources, such as demonstrations either from human or another RL agent [2,11], or expert's guidance, etc.

The optimal policy is determined by the reward function, small transformations of the reward function might cause intractable problems [15]. Ng. et al. proved that by following the potential-based reward shaping, the optimal policy will be invariant [13]. The original reward function R is augmented by another reward function F, shown in Eq. 1 and if and only if F is the subtraction between a function ϕ of the next state s' and the current state s then the optimal policy will keep unchanged. The function ϕ is called potential function. Examples of good potential function could be Manhattan distance in navigation tasks or pre-trained state value functions, etc.

$$\begin{aligned} R'(s,a,s') &= R(s,a,s') + F(s,a,s') \\ &= R(s,a,s') + \phi(s') - \phi(s) \end{aligned} \tag{1}$$

Brys et al. extracted potential function from demonstrations by checking if agent's state-action pairs are in demonstrations or not and apply it to Cart Pole and Mario [2]. Hussein et al. trained a neural network from demonstrations as the potential function and added it to the original reward function of DQN in grid navigation tasks[12]. Grzes provided more insights and analysis for potential-based reward shaping and extended it to multi-agent RL scenario [7]. While most previous methods have focused on extracting potential functions from expert demonstrations, we investigate whether potential functions can also be extracted from a search. In our case, we use the distance function which is provided by the A* search algorithm as the potential function.

2.2 Sokoban

Sokoban is a challenging puzzle game and has been proved to be PSPACE-complete [3] and NP-hard [4] problem. It also plays an important role in benchmarking RL agents. Many models are proposed to solve Sokoban. Both model-based methods [9,10,21], as well as model-free methods can reach competitive performance [8]. Curriculum learning has been used to solve a difficult Sokoban instance [6]. The works mentioned above try to solve Sokoban using special-designed models, while we are focusing on using general reward shaping techniques to speed up the learning.

Fine-tuning pre-trained models is helpful in accelerating learning in Sokoban [22]. Reward shaping was applied to a single simple Sokoban instance by interacting with human experts in [16] to speed up the learning. In our work, we demonstrate potential-based reward shaping over many Sokoban instances range from 1-box to 3-boxes, where no human expert involved.

3 Methods

In this section, we will explain the methods and techniques that we used. We report the problem model, the heuristic that was used in A*, and details about the reward shaping.

3.1 Reinforcement Learning

MDP is short for Markov Decision Process, and it models decision making problems into a 4-tuple, $\langle S, A, R(s,a,s'), P(s,a,s')\rangle$. In our paper, we follow the MDP notation proposed in [19]. S is a set of states s called the state space, it will be different states in Sokoban instance in our case. A is a set of actions a called the action space, in our case it will contain all actions that the agent can take (no operation, going up/down/left/right). $R(s,a,s')$ is a reward function that determines immediate rewards that the agent will get after performs an action a which leads the agent from the current state s to the next state s'. $P(s,a,s')$ is the probability that action a leads the agent from the current state s to the next state s'. Reinforcement learning methods solve MDP using data$(s,a,s',R(s,a,s'))$ collected by interacting with the environment to train a policy aims to maximize the accumulated reward shown in Eq. 2,

$$R = \sum_{t=0}^{\infty} \gamma^t r_t \tag{2}$$

where $\gamma \in [0,1]$ is the discount factor and r_t is the immediate reward the agent gets in time step t.

We use the RL algorithm A2C to train the agent to learn to solve Sokoban. The policy is represented by a neural network and the architecture of the neural network we are using for experiments is the same as the architecture used in [22], which consists of 3 convolutional layers and 2 fully-connected layers. All hyperparameters of A2C are also kept the same as described in [22]. More details can be found in the Appendix B.

3.2 A* Heuristics

A* is a heuristic search algorithm, it extends the Dijkstra's algorithm by adding heuristics. The heuristic we used in our case is the overall Manhattan distance between untargeted boxes and goals[5], formula shown in Eq 3.

$$h(s) = \sum_{b \in B, t \in T} (|x_b - x_t| + |y_b - y_t|) \tag{3}$$

, where (x_b, y_b) is the location of boxes while (x_t, y_t) is the location of targets, h is the heuristic for the current state s. B is all boxes which are not on targets

[5] The implementation we are using is from `https://github.com/KnightofLuna/sokoban-solver`

yet and T is all targets where there are no boxes on. If a Sokoban instance is solvable, A* will return the solution otherwise it will return nothing. As such, it could also be used to check the solvability of a Sokoban instance.

3.3 Reward Shaping

Values of potential functions of states should be higher if states are 'better' and vice versa. For this reason the **minus** of the distance function is used as the potential function in our case. The distance function will take the current state as input, and output how many steps the agent needs to take towards the goal state.

The shaped reward function will be(shown in Eq. 4):

$$\begin{aligned} R'(s,a,s') &= R(s,a,s') + F(s,a,s') \\ &= R(s,a,s') + \phi(s') - \phi(s) \\ &= R(s,a,s') - d(s') + d(s) \end{aligned} \tag{4}$$

where R' is the new shaped reward while R is the original extrinsic reward provided by the environment. s is the current state while s' is the next state, and a is the action that leads s to s'. ϕ is the potential function and $d(s)$ is the distance function from the current state s to the goal state. In our case, we use the A* search algorithm[6] to provide the distance information.

In Sokoban, some actions can lead to unsolvable situations. An example is shown in Fig. 2. A box is pushed into the corner and it is not possible to pull it back. The instance has become completely unsolvable. Then a natural question is what distance we should assign to states which are unsolvable. The algorithm, however, can still get some rewards by learning sub-optimal policies, such as pushing one of the boxes onto one of the targets. In order not to break the sub-optimal policy invariance, we don't shape the reward function and just keep the original reward function after the instance has become unsolvable. To conclude, our shaped reward function is shown in Eq. 5. It is important to note, when in the step that leads the agent to an unsolvable situation, we make $d(s')$ one step further than $d(s)$, which will be $d(s) - 1$. Although $d(s')$ does not exist, it is reached by taking one step further from its predecessor state s. We found this strategy performs better than other methods we tried such as giving a large penalty or not giving any penalty, etc.

$$F(s,a,s') = \begin{cases} (\phi(s) - 1) - \phi(s) = -1 & \text{if } s \text{ is solvable and } s' \text{ is unsolvable} \\ 0 & \text{if both } s' \text{ and } s \text{ are unsolvable} \\ \phi(s') - \phi(s) = -d(s') + d(s) & \text{otherwise} \end{cases} \tag{5}$$

[6] https://en.wikipedia.org/wiki/A*_search_algorithm

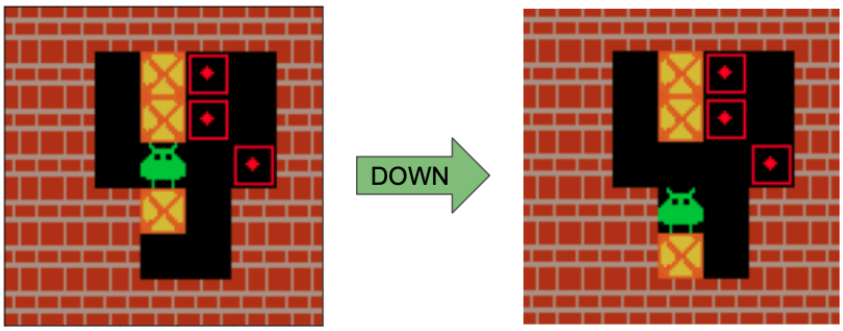


Fig. 2: How an unsolvable situation happens.

4 Experiments

The agent is evaluated every 1,000 environment steps on 20 randomly selected instances. We use 100 * 1-box instances, 100 * 2-boxes instances and 60 * 3-boxes instances(since more boxes are more expensive, we use 60 instead of 100). The results shown are averaged over 5 runs with different random seeds.

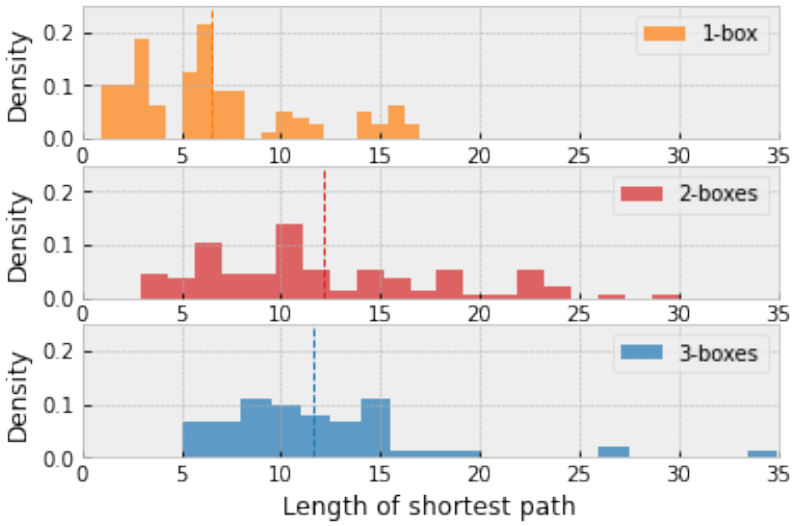


Fig. 3: The shortest path of different Sokoban instances, dash lines are means. Top: solutions of 1-box instances, and the mean of solutions is 6.52. Mid: solutions of 2-boxes instances, and the mean of solutions is 12.14. Bottom: solutions of 3-boxes instances, and the mean of solutions is 11.62.

The length of the shortest path of an instance could indicate the difficulty of the instance. Fig. 3 shows the distribution of shortest paths of instances we are using; as expected the more boxes, the longer the shortest path.

The learning results on 1-box instances is shown in Fig. 4. Even without reward shaping, the agent can quickly learn to master given instances within 60k environment steps. From the top subplot in Fig. 3 we see that, solutions of 1-box instances are mostly shorter than 10. This also indicates that RL is able to solve simple sparse-reward problems. By adding reward shaping, the agent is about four times faster than learning from scratch. Both learning with reward shaping and learning from scratch are able to solve given instances within the given steps.

Learning on 2-box and 3-box instances is more difficult than learning on 1-box instances. Solutions of multiple-box instances are generally longer than solutions of single-box instances. Reinforcement learning from scratch almost learns nothing on 2-box and 3-box instances. In Fig.5, we can see that learning with reward shaping performs better by a large margin, the agent is able to solve given instances within 50k environment steps, while learning from scratch only reaches a solved ratio of around 0.2. This again demonstrates that RL is not good at solving sparse-reward problems.

Fig. 4: Learning over 100 * 1-box instances.

5 Discussion and Conclusion

In this work, we showed that the distance function can be used as potential function in potential-based reward shaping to further speed up the learning in

Fig. 5: Learning over 100 * 2-boxes instances(left) and 60 * 3-boxes instances(right).

Sokoban. Meanwhile, the distance function can also be used as potential function in grid-world navigation tasks, since grid-world navigation tasks can be treated as special types of Sokoban where there are no needs for pushing boxes but only moving the agent to the target. Our experiments showed that abilities of solving multiple instances can be quickly learned and compressed into a single behavior policy, which can be seen as the first step towards training a general policy which is able to solve unseen Sokoban instances quickly. For instance, if the agent is exposed to a Sokoban generator for training and the goal is to train the agent to be able to solve new unseen instances. Then learning with reward shaping will be way faster than learning from scratch to reach this 'final' goal.

A limitation of our approach is that since we are using search algorithms to provide the distance function, scalability is limited. For more difficult instances, search algorithms can not find solutions within a reasonable time, and the reward shaping we did in this work will not be usable. In the future, it would be interesting to work on scalable heuristic functions to use as potential functions in Sokoban. On the other hand, as we mentioned, our methods can be firstly used to train a baseline agent quickly, thus a interesting future work can also be to find possibility to reuse or transfer the trained neural networks for further training or tasks.

Acknowledgement

The financial support to Zhao Yang is from the China Scholarship Council(CSC). Computation support is from ALICE/Leiden and the DSLab. The authors thank Michiel van der Meer, Hui Wang, Matthias Müller-Brockhausen and all members from the Leiden Reinforcement Learning Group for helpful discussions. Especially thank Thomas Moerland for detailed feedback.

References

1. Badia, A.P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z.D., Blundell, C.: Agent57: Outperforming the atari human benchmark. In: International Conference on Machine Learning. pp. 507–517. PMLR (2020)
2. Brys, T., Harutyunyan, A., Suay, H.B., Chernova, S., Taylor, M.E., Nowé, A.: Reinforcement learning from demonstration through shaping. In: Twenty-fourth international joint conference on artificial intelligence (2015)
3. Culberson, J.: Sokoban is pspace-complete (1997)
4. Dor, D., Zwick, U.: Sokoban and other motion planning problems. Computational Geometry **13**(4), 215–228 (1999)
5. Dorigo, M., Colombetti, M.: Robot shaping: Developing autonomous agents through learning. Artificial intelligence **71**(2), 321–370 (1994)
6. Feng, D., Gomes, C.P., Selman, B.: A novel automated curriculum strategy to solve hard sokoban planning instances. Advances in Neural Information Processing Systems **33**, 3141–3152 (2020)
7. Grzes, M.: Reward shaping in episodic reinforcement learning (2017)
8. Guez, A., Mirza, M., Gregor, K., Kabra, R., Racanière, S., Weber, T., Raposo, D., Santoro, A., Orseau, L., Eccles, T., et al.: An investigation of model-free planning. In: International Conference on Machine Learning. pp. 2464–2473. PMLR (2019)
9. Guez, A., Weber, T., Antonoglou, I., Simonyan, K., Vinyals, O., Wierstra, D., Munos, R., Silver, D.: Learning to search with mctsnets. In: International conference on machine learning. pp. 1822–1831. PMLR (2018)
10. Hamrick, J.B., Friesen, A.L., Behbahani, F., Guez, A., Viola, F., Witherspoon, S., Anthony, T., Buesing, L.H., Veličković, P., Weber, T.: On the role of planning in model-based deep reinforcement learning. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=IrM64DGB21`
11. Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al.: Deep q-learning from demonstrations. In: Thirty-second AAAI conference on artificial intelligence (2018)
12. Hussein, A., Elyan, E., Gaber, M.M., Jayne, C.: Deep reward shaping from demonstrations. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 510–517. IEEE (2017)
13. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: Theory and application to reward shaping. In: In Proceedings of the Sixteenth International Conference on Machine Learning. pp. 278–287. Morgan Kaufmann (1999)
14. Plaat, A.: Learning to Play: Reinforcement Learning and Games. Springer Verlag, Heidelberg, See `https://learningtoplay.net` (2020)
15. Randløv, J., Alstrøm, P.: Learning to drive a bicycle using reinforcement learning and shaping. In: ICML. vol. 98, pp. 463–471. Citeseer (1998)
16. Raza, S.A., Johnston, B., Williams, M.A.: Reward from demonstration in interactive reinforcement learning. In: The Twenty-Ninth International Flairs Conference (2016)
17. Schrader, M.P.B.: gym-sokoban. `https://github.com/mpSchrader/gym-sokoban` (2018)
18. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature **529**(7587), 484–489 (2016)

19. Sutton, R.S., Barto, A.G.: Reinforcement learning, An Introduction, Second Edition. MIT Press (2018)
20. Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature **575**(7782), 350–354 (2019)
21. Weber, T., Racanière, S., Reichert, D.P., Buesing, L., Guez, A., Rezende, D.J., Badia, A.P., Vinyals, O., Heess, N., Li, Y., et al.: Imagination-augmented agents for deep reinforcement learning. arXiv preprint arXiv:1707.06203 (2017)
22. Yang, Z., Preuss, M., Plaat, A.: Transfer learning and curriculum learning in sokoban. arXiv preprint arXiv:2105.11702 (2021)

A Environment

The environment we are using is from the implementation [17] with slight modification. The rewards of the environment shown in the Tab. 1. Solving the instance by pushing all boxes onto targets returns 10.0; pushing one box onto a target gets 1.0 and pushing it off gets -1.0; in order to incentivize the agent solve the instance quickly, an -0.1 is given for each step the agent makes.

Table 1: Rewards in the environment.

actions	reward
push all boxes on targets	10.0
push one box onto target	1.0
push one box onto target	-1.0
each step	-0.1

B Neural Network Details

The model we are using contains three convolutional layers with kernel size 8x8, 4x4, 3x3, strides of 4, 2, 1, and number of output channels 32, 64, 64. Then followed by a fully connected layer with 512 units. In the end, the outputs are fed into two heads: outputting the policy logits and the state value. `ReLU` is used as the activation function after each layer and `RMSprop` is the optimizer we used. The input is pixel image gets from the environment directly, which is 3x80x80.

Table 2: Hyper-parameters of the neural network and training.

learning rate	$7 \cdot 10^{-4}$
gamma	0.99
entropy coef	0.1
value loss coef	0.5
eps	10^{-5}
alpha	0.99
rollout storage size	5
No. of environments for collecting trajectories	30

C Overall Training Loop

Algorithm 1: Overall RL training loop

Initialization: policy π, number of training steps N, environment `env`
$s \leftarrow$ `env.reset()`;
while $n < N$ **do**
  $a \leftarrow \pi(s)$;
  s', r, $\leftarrow$ `env.step`(a);
  `/* calculate the potential value under different situations. */`
  if *s and s' are solvable* **then**
    $f \leftarrow -d(s') + d(s)$; `/* f is the potential value. */`
  else if *s is solvable and s' is not solvable* **then**
    $f \leftarrow -d(s) - 1$;
  else if *s, s' are not solvable* **then**
    $f \leftarrow 0$;
  $r' \leftarrow r + f$; `/* Reshape the reward. */`
  execute A2C update on π using the shaped reward r';
  $n \leftarrow n + 1$;
end
return π ;

Distinguishing Commercial from Editorial Content in News

Timo Kats[1[0000−0003−1650−1814]], Peter van der Putten[1[0000−0002−6507−6896]], and Jasper Schelling[2[0000−0002−9995−1505]]

[1] Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
t.p.a.kats@liacs.leidenuniv.nl, p.w.h.van.der.putten@liacs.leidenuniv.nl
[2] Stichting ACED, Amsterdam, The Netherlands
jasper@aced.site

Abstract. How can we distinguish commercial from editorial content in news, or more specifically, differentiate between advertorials and regular news articles? An advertorial is a commercial message written and formatted as an article, making it harder for readers to recognize these as advertising, despite the use of disclaimers. In our research we aim to differentiate the two using a machine learning model, and a lexicon derived from it. This was accomplished by scraping 1.000 articles and 1.000 advertorials from four different Dutch news sources and classifying these based on textual features. With this setup our most successful machine learning model had an accuracy of just over 90%. To generate additional insights into differences between news and advertorial language, we also analyzed model coefficients and explored the corpus through co-occurrence networks and t-SNE graphs.

Keywords: advertorials · NLP · t-SNE · co-occurrence networks

1 Introduction

In journalism it is best practice to clearly distinguish between editorial and sponsored commercial content. This is referred to as the 'separation of church and state' in media [2]. However, some forms of advertising have made this separation less clear to readers and therefore threaten this principle.

An example of this is the advertorial, which is commercial content in the form of an article. Advertorials are an example of what marketers call 'native advertising'. In fact, advertorials are so much like articles, that despite using disclaimers and different layouts most readers don't notice the difference. In a study conducted by the university of Georgia only 8% of readers recognized advertorials as commercial content [16]. As a result of this, advertorials have made the separation of church and state in the news less clear.

That's why this research aims to differentiate articles and advertorials using machine learning. We would like to answer two research questions. Firstly, to what extent can we differentiate commercial and editorial content by a using machine learning model, and a lexicon derived from this? Secondly, can we use

AI and machine learning to better understand the difference between commercial and editorial language?

It's important to note that the separation of commercial and editorial content is hotly debated in journalism and the society at large. Yet, to our knowledge a machine learning based perspective to identify advertorials and commercial messaging was not part of this debate yet. By doing this we not only hope to answer our research questions, but also showcase how machine learning can be a solution in the debate surrounding the usage of advertorials. This research has been carried out in the context of the Reverb Channel program [12], a data driven exploration of our networked news culture that aims to reverse the sometimes questionable role of AI in digital media, by using it to investigate topics such as framing, polarization and ideology spaces.[3]

The remainder of this paper is structured as follows. Section 2 provides more background and related work. Section 3 explains the process of acquiring our data, followed by sections on our classification approach, and on our exploratory co-occurrence network based approach to increase the insight into how language differs across advertorials and news. Section 6 concludes the paper.

2 Background and Related Work

Even though we are not aware of any other research to leverage machine learning to distinguish advertorials from editorial content, the discussion around the usage of advertorials and commercial content in general is broader than this research alone, and has been debated widely in journalism and marketing. In this section we discuss some of this background context.

2.1 The change of journalism's business model in the digital age.

The rise of the internet has had a lot of effect on journalism. It opened up a whole new channel for news content, but it also it negatively impacted circulation and advertising revenue for traditional news channels. For example, US weekend circulation of newspapers declined from 59.4 million (2000) to est. 25.8 million (2020), revenue from advertising declined from 48.7 billion (2000) to est. 8.8 billion (2020), whilst revenue from circulation remained relatively stable (10.5 (2000) to 11.1 (2020)), and the share of advertising revenue increased from 17% (2011) to 39% (2020) [10]. So despite drastic drops in circulation, companies were able to protect circulation income, but advertising revenues dropped dramatically. These developments altered the business model of journalism significantly, and drove publishers to find new sources of advertising revenue, such increased usage of advertorials and other forms of sponsored content.

2.2 Disguise, deception and disclosure in advertorials.

As discussed, whilst in journalism the distinction between editorial and sponsored commercial content is a key principle, this is challenged by advertorials in

[3] `https://www.aced.site/en/programmes/reverb-channel`

practice as readers have a hard time differentiating these from editorial content, despite the use of labels and disclaimers [2].

Advertorials can be both deceptive and effective. As a classical example, in 1989 the R.J Reynolds Tobacco Company had settled charges with the FTC on that it had made false and misleading claims in an advertorial on health effects of smoking, titled 'Of cigarettes and science'. Wilkinson et al. subsequently ran a test and over a quarter of participants thought the article was editorial content, not commercial [15].

In another study by Kim et al., the use of an advertorial over a standard advertisement increased the relevance of and attention to the message, and message and elaboration and recall. It made no difference whether the advertorials were labeled as such, and over two thirds of subjects exposed to labeled advertorials were not able to recall whether these advertorials were labeled or not [7].

As mentioned in the introduction, in another study by the University of Georgia only 8% of readers recognized advertorials as commercial content [16]. In their study, the use of disclaimers did have a positive impact on recognizing the text as commercial, with best effects for placement of disclaimers in the middle or the bottom, and explicit use of words such as 'advertizing' and 'sponsored'. Also Krouwer et al. found that small changes, such as the location of a disclaimer, significantly impacts the recognizability for readers [8]. Apart from readers not noticing labeling, advertorials often violate guidelines for labeling, formatting and content [1].

To provide perhaps a somewhat more positive view on advertorials, in a survey by Reijmersdal et al. of subscribers of Dutch magazines, when asked explicitly only 12% of respondents thought advertorials are deceptive [11].

The more established newspapers and magazines will make more of a serious effort to make it known that certain content is sponsored, and writers producing advertorials are kept separate from the editorial teams. But is that sufficient, also when taking the proliferation of new digital media titles and the ongoing pressure to increase advertizing revenues into account, and norms are shifting towards further integration between editorial and commercial teams and objectives [3]?

The results above may vary but in our opinion this is clearly not sufficient. The ability to disguise content, willingly or unwillingly, and the probability that advertorials are not recognized as such even if properly labelled is significant. Marketers call it native advertizing for a reason.

The risk of mistaking commercial content for objective editorial content is somewhat obvious, but note there can be an opposite detrimental effect as well. For instance, Iversen at al. observed that exposure to native political ads reduced the public's trust in political news [5].

2.3 The usage of lexicons in classifying text

As mentioned, we aim to create a classification model and lexicon that distinguishes editorial from commercial language. Whilst text classification models are used abundantly in NLP research, we are also looking to distribute our artifacts

to journalists and other non-technical audiences. In domains such as social science lexicons are often used, for common tasks such as sentiment analysis [13] or more specific tasks, such as detecting moral foundations in ethical reasoning [4, 14]. Lexicons can be handcrafted or created through linguistic analysis, and typically include keywords that indicate a particular class, potentially including a weight.

We were not able to identify prior work that uses machine learning, handcrafted or trained lexicons to differentiate advertorials from editorial content. A different, yet relevant related work is the study by Zhou, who uses genre analysis to characterize the general structure and linguistic characteristics of advertorials, using mostly manual analysis and interpretation [17].

3 Data Acquisition

In order to make a model that answers the research questions mentioned earlier we have created a data set with advertorials and regular news articles. The Reverb Channel corpus contains millions of articles [12], but no advertorials, hence we had to acquire our own data for this research. In this section we explain this process and showcase the data set that we acquired. For full details we refer to [6].

3.1 Scraping the data

The data for this research had to be scraped directly from news sources using web crawlers. For our research we used Python and the BeautifulSoup library. With this set up we made a URL-scraper and a web-scraper for every news source. We first collected the URLs from the pages we wanted to scrape data from and thereafter use those URLs to collect all the data we needed with the web-scraper. We also carried out additional cleaning and transformation, such as removal of all commas, translation of any HTML to flat text where needed, and lowercasing of all text.[4]

3.2 Resulting data set

The data set that we acquired with this method has 2000 entries in total, about half of these entries are advertorials (see Figure 1). These entries are roughly equally distributed over four different news sources. These news sources are (online-only news) Nu.nl, (politically conservative) Telegraaf, (politically progressive) NRC, and (business publication) De Ondernemer. By including these four different news sources with roughly equal number of documents in the data set we strive to create an unbiased data set that is representative of the Dutch media landscape as a whole.

[4] Source code, the lexicon and other deliverables can be found at `https://github.com/TimoKats/research_distinguishing_commercial_and_editorial_content`

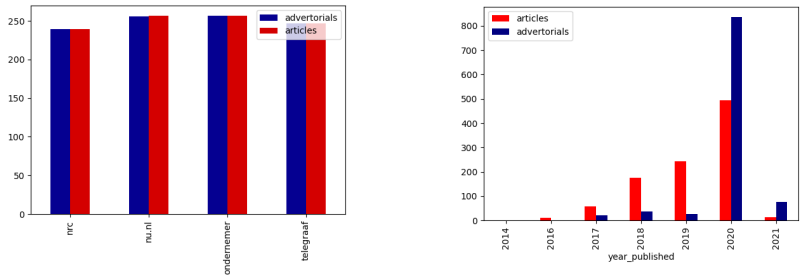


Fig. 1: Metadata from the acquired data set

4 Distinguishing Advertorials from Regular Articles with Classification Models

With this corpus we developed classification models and a corresponding lexicon, which also gave us some first insights into differences between the language used in advertorials versus news.

4.1 Experimental set up

In terms of cleaning the data, we first removed potential leakers. Leaking variables in our model refer to words that trigger the model whilst being unique to our data set and media covered, for example sponsor names and disclaimers. To further lower the risk of leakage, we excluded the title and focused on the main text. Furthermore, we experimented with regular bag of words (BoW) as well as TFIDF weighted BoW, the removal of stop words and the number of features. Obviously, we could have easily obtained classifiers with near perfect performance, for instance by including disclaimer texts, but we were primarily interested in models that could distinguish commercial from editorial language.

For modeling, we selected a diverse set of classification methods to experiment with: SVM (default with rbf kernel), linearSVC, decision tree, random forest, k-NN, SGD and naive bayes. We restricted ourselves to these more classical methods as opposed to deep learning methods such as BERT, given that our data sets are relatively small, and interpretability of the results is key, for instance to iteratively identify leakers and get more insights into the difference between text types.

To simplify the approach, we aim to find the best performing model (incl. parameter optimization) through narrowing down the search as the experiment progresses, taking the best performing preliminary results and continuing to optimize it. A limitation of such an approach is that the estimate of final accuracy may be somewhat optimistic given the sequential nature of the experiments (manual overfitting), but a full multidimensional experimental set up was too

representation	learning model	accuracy	f1 score	auc
bag of words	svm	0.85±0.04	0.85±0.05	0.93±0.04
bag of words	linearSVC	0.84±0.05	0.84±0.06	0.9±0.05
bag of words	decisionTree	0.78±0.08	0.78±0.08	0.79±0.07
bag of words	randomForest	0.88±0.06	0.89±0.07	0.94±0.05
bag of words	k-NN	0.57±0.14	0.63±0.12	0.58±0.16
bag of words	SGD	0.87±0.07	0.86±0.08	0.93±0.05
bag of words	naiveBayes	0.76±0.11	0.77±0.09	0.76±0.11
tfidf	svm	0.89±0.05	0.89±0.05	0.94±0.04
tfidf	linearSVC	0.91±0.05	0.91±0.05	0.95±0.03
tfidf	decisionTree	0.78±0.07	0.79±0.07	0.8±0.07
tfidf	randomForest	0.88±0.07	0.89±0.06	0.94±0.05
tfidf	k-NN	0.51±0.03	0.64±0.02	0.51±0.05
tfidf	SGD	0.9±0.05	0.9±0.06	0.95±0.04
tfidf	naiveBayes	0.76±0.09	0.76±0.08	0.76±0.1

Table 1: Cross-validation accuracy with removal of stop words

computationally expensive, and the scarcity of advertorials limited the use of an additional hold out test set. This could be addressed in future work.

For SVM, SGD and linearSVC we increased the maximum amount of iterations to 5000 and for decision tree and random forest we set the max depth to "none". In terms of evaluation we ran 10-fold cross validation to test various algorithms and parameters, as well as a cross domain test set up where one medium is used as the test set, and models are trained on the other media. The metrics that we evaluate our results with are accuracy, f1 score and AUC.

4.2 Results

In a first set of experiments we benchmark the performance of all algorithms across regular and TF-IDF weighted BoW representations. Table 1 shows the results, with stop words removed; the results with stop words included were very similar. TF-IDF typically outperformed regular BoW so the remainder of the experiments was carried out with TF-IDF, with stop word filtering.

The results of the cross domain testing experiment can be found in Table 2. The best results were obtained with SVM, linearSVC, random forest and SGD, closely followed by decision trees and naive bayes, and k-NN scored poorly, probably due to high dimensionality. Top scoring results were close, but SVM scored best, so we decided to continue the experiments with this method. In terms of media, NRC scored best, followed by Nu.nl and Telegraaf, and Ondernemer scoring substantially worse, which may be due to the fact that in the business to business domains editorial and commercial content is more similar.

We also ran a structured experiment where we gradually increased the number of features that made clear that at 5000 features performance more or less stabilizes (results omitted for brevity), and we ran a series of tests to study the impact of tweaking the various SVM parameters (Table 3).

	SVM	Linear SVC	Decision Tree	Random Forest	k-NN	SGD	Naive Bayes	
Nu.nl	0.84	0.84	0.72	0.83	0.52	0.84	0.72	0.76 ± 0.12
NRC	0.93	0.95	0.75	0.82	0.52	0.95	0.81	0.82 ± 0.15
Ondernemer	0.76	0.68	0.54	0.55	0.51	0.65	0.56	0.61 ± 0.09
Telegraaf	0.85	0.84	0.66	0.84	0.46	0.83	0.73	0.74 ± 0.14
	0.85 ±0.06	0.83 ±0.10	0.67 ±0.08	0.76 ±0.12	0.5 ±0.02	0.82 ±0.11	0.71 ±0.09	

Table 2: Cross-domain testing results (test set in rows, trained on other media, metric is accuracy)

To further validate the cross domain results, we also trained and tested models with data from just one medium each, and created t-SNE graphs (Figure 2, along with the corresponding accuracies). t-SNE graphs [9] are a way to represent multi-dimensional data (in our case a 5000 dimensions) in a two-dimensional scatter plot. For our experiment, we ran the t-SNE graph with a perplexity of 30, a maximum number of iterations of 1000 and a random state of 2. In other words, apart from the random state only the default parameter values.
Using this method we can visualize how well the classes can be separated based on the available data, making it possible to visualize the separation of church and state in our experiment. The ranking of various media are consistent with the cross domain results, with NRC displaying the clearest separation and Ondernemer the worst.

After completing the experimental process explained earlier we found that the model explained in Table 4 gave us the best results (all other parameters are defaults). So we used this model to derive a lexicon by training a model on all data and using this model's feature terms and weights. This is useful, even though it serves the same purpose as our model, because it can be published without publishing the data as well, which we are not able to do because of copyright issues, and it can be consumed more easily by a broad non technical audience such as journalists and social scientists. [5]

Using a linear kernel means that the separating hyperplane is defined in the original input space, hence we can interpret the weights of the model as term weights in a lexicon. Users can make very simple use of the lexicon, just by counting the occurrence of negative and positive words (with zero as threshold) or approximate the original model closer, for example by calculating a score by multiplying frequency of the terms with the term weights and summing the results. Figure 3 shows the distribution of the scores for the full corpus, calculated with the latter approach. One can clearly see two more or less normal distributions representing the advertorials and regular articles.

Inspection of these feature coefficients also provides further insight into differences in language use between classes. In Figure 4 we have listed the features

[5] The lexicon is published at `https://github.com/TimoKats/research_distinguishing_commercial_and_editorial_content`

kernel	decision function	accuracy	f1_score	roc_auc
linear	ovo	0.9029±0.0559	0.9003±0.0581	0.9495±0.0341
linear	ovr	0.9029±0.0559	0.9003±0.0581	0.9495±0.0341
poly	ovo	0.8094±0.0774	0.7755±0.1016	0.9167±0.0412
poly	ovr	0.8094±0.0774	0.7755±0.1016	0.9167±0.0412
rbf	ovo	0.8999±0.0579	0.8989±0.0583	0.9438±0.0393
rbf	ovr	0.8999±0.0579	0.8989±0.0583	0.9438±0.0393
sigmoid	ovo	0.9009±0.0563	0.8986±0.0585	0.9498±0.0339
sigmoid	ovr	0.9009±0.0563	0.8986±0.0585	0.9498±0.0339

Table 3: The effect of tweaking the parameters with svm

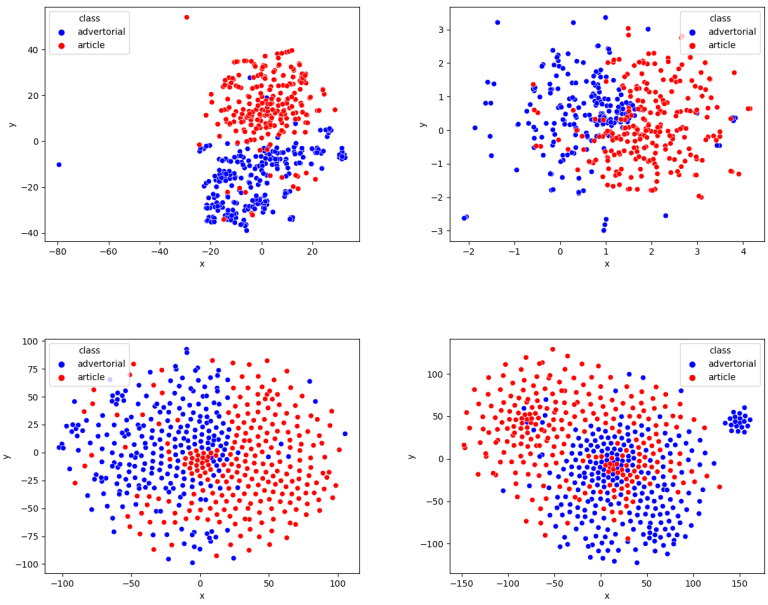


Fig. 2: t-SNE plots for NRC (95%), Nu.nl (92%), Telegraaf (91%) and Ondernemer (85%)

learning model	features	text representation	kernel	max iter	accuracy
svm	5000	tf-idf	linear	5000	0.9029±0.0559

Table 4: Settings from the final model.

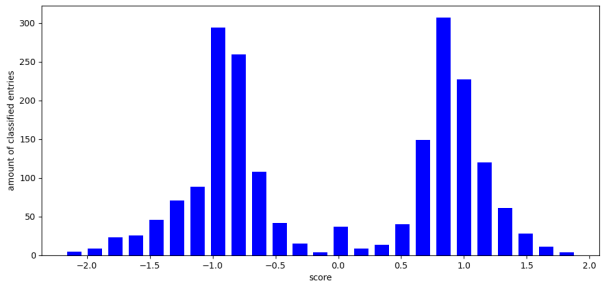


Fig. 3: The distribution of the predicted scores from the entries using the lexicon.

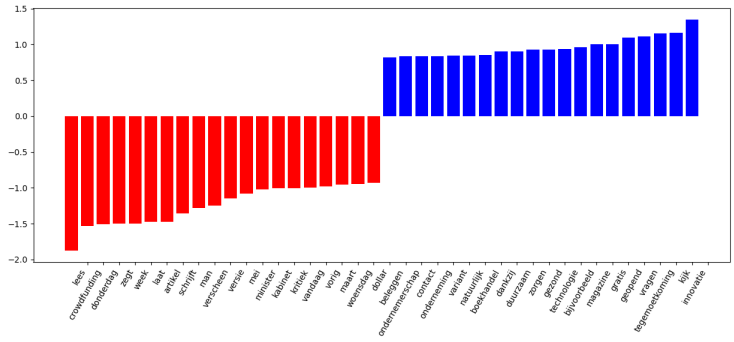


Fig. 4: Coefficients with highest absolute values for regular articles (left) and advertorials (right)

with the highest absolute values for regular articles and advertorials. As can be seen, the difference isn't just a matter of topics (f.i. 'cabinet', 'minister' for news, 'investing', 'enterprise', 'technology', 'innovation' for advertorials), but also a matter of how these topics are being talked about. In regular articles, indications of time (days, months etc) and attribution ('writes', 'says', 'appeared') score high, whereas high scoring features for advertorials include adjectives such as 'free', 'healthy' and 'sustainable', perhaps highlighting the benefits of products and services. Coefficients for the full 5000 features can be downloaded with the lexicon, and we also investigated by training models on media separately to understand differences between publications.

5 Exploring the Corpus with Co-occurrence Networks

Results such as feature importance already provided us with some insights into how language differs between advertorials and regular articles, but to delve

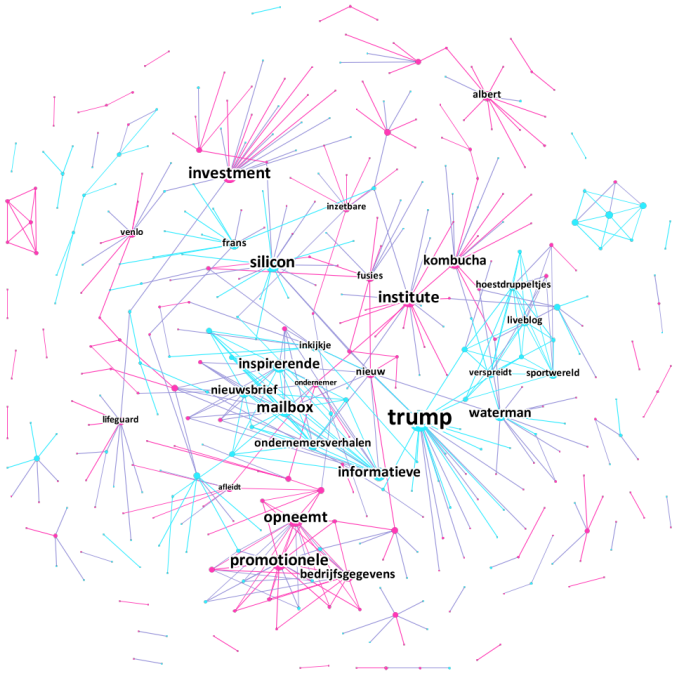


Fig. 5: Overview of the co-occurence network. Complete version can be found at: `timokats.github.io/network/`

deeper in a more exploratory fashion we have also created co-occurrence networks (see Figure 5).

The nodes in this network are the terms from the lexicon, blue nodes are the editorial terms and the red nodes are the commercial terms (negative and positive coefficients). The size of the node is related to its degree. The edges represent the connection between two terms in the data set. We calculated this based on how often the two terms appear in the same sentence as a percentage. So for example, in our data set every time the term 'artificial' appears in a sentence, 75% of the time that sentence also has the word 'intelligence'. Thus, there's a directed edge from 'artificial' to 'intelligence'. For visualization we show all nodes with edges exceeding a minimum threshold, or likewise this could be seen as an undirected graph where the weight is the lowest value for the two terms.

By exploring the co-occurrence network certain things about both our results and data become apparent. First, the fact that some of our data centers around subjects that were very prevalent in 2020 (like the US elections and the covid-19 pandemic), resulting in a time frame bias towards 2020, because in fu-

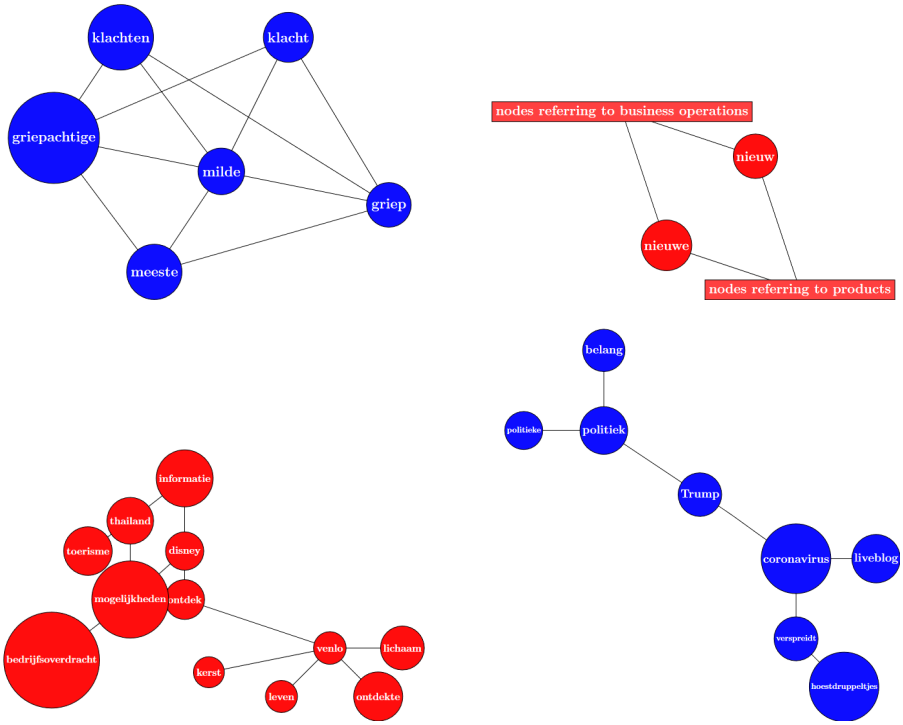


Fig. 6: Commercial and editorial sub graphs.

ture implementations of our model and/or lexicon these subjects may be less prevalent.

Second, it has also given us more insight into the structure of commercial language and how it's different from editorial language. For example, commercial language in our network has two large clusters (one related to goods and one related services). These clusters are linked by the terms 'nieuw' (new) and 'nieuwe' (new). For our editorial clusters we for example found a cluster related to covid symptoms, which showcases the time frame bias mentioned earlier. Through using a co-occurrence graph we can find patterns/clusters like these and gain more insight into our data and results. An overview of some important findings in our graph can be found in Figure 6.

6 Conclusion

This research aims to differentiate commercial and editorial content, and more specifically, advertorials from regular articles, and our main research questions are the following. To what extent can we differentiate advertorials and articles by using machine learning? And can we use machine learning and a data driven

approach to better understand the difference between commercial and editorial language?

We answered the first question by developing a range of models for various media, and deriving a lexicon from it. The best models perform with over 90 per cent accuracy, and as mentioned this is an optimistic estimate and performance clearly varies by medium and set up. Further insight is provided by highlighting the differences of performance across media, with business-to-business medium Ondernemer scoring lowest, which could make sense given similarities in jargon. Feature importance analysis and co-occurrence graphs provided further insight into differences in language, both from a topic perspective, as well as how these topics were being spoken about.

Our research has some known limitations. In particular the size of the data set (of just 2000 entries) could be increased in future work, including a wider set of media and longer time frames. A key challenge here to overcome is that that particularly advertorials are not always available for extended periods of time. It may also be interesting to expand the scope to other major languages and other forms of native advertising. We also plan to engage with the general public, journalists as well as marketers, using the results of this research to raise awareness and trigger debate and discussion.

Despite some of it's limitations we think our research can serve as an example to put the problem on the agenda, provide insight into it, and illustrate the potential of using machine learning for differentiating commercial and editorial content. Moreover, it also showcases how machine learning and AI can be a solution, not just a problem, in society and the modern digital media landscape.

References

1. Cameron, G.T., Ju-Pak, K.H., Kim, B.H.: Advertorials in magazines: Current use and compliance with industry guidelines. Journalism & Mass Communication Quarterly **73**(3), 722–733 (1996)
2. Conill, R.F.: Camouflaging church as state: An exploratory study of journalism's native advertising. Journalism Studies **17**(7), 904–914 (2016)
3. Cornia, A., Sehl, A., Nielsen, R.K.: 'We no longer live in a time of separation': A comparative analysis of how editorial and commercial integration became a norm. Journalism **21**(2), 172–190 (2020)
4. Graham, J., Haidt, J., Nosek, B.A.: Liberals and conservatives rely on different sets of moral foundations. Journal of personality and social psychology **96**(5), 1029 (2009)
5. Iversen, M.H., Knudsen, E.: When politicians go native: The consequences of political native advertising for citizens' trust in news. Journalism **20**(7), 961–978 (2019)
6. Kats, T.: Differentiating commercial and editorial content (2021), BSc thesis, Leiden University
7. Kim, B.H., Pasadeos, Y., Barban, A.: On the deceptive effectiveness of labeled and unlabeled advertorial formats. Mass Communication and Society **4**(3), 265–281 (2001)

8. Krouwer, S., Poels, K., Paulussen, S.: To disguise or to disclose? the influence of disclosure recognition and brand presence on readers' responses toward native advertisements in online news media. Journal of Interactive Advertising **17**, 00–00 (10 2017)
9. van der Maaten, L., Hinton, G.: Viualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (11 2008)
10. Pew Research Center: State of the media: Newspapers. `https://www.pewresearch.org/journalism/fact-sheet/newspapers/`, Online: accessed: 2021-09-07
11. van Reijmersdal, E., Neijens, P., Smit, E.: Readers' reactions to mixtures of advertising and editorial content in magazines. Journal of Current Issues & Research in Advertising **27**(2), 39–53 (2005)
12. Schelling, J., van Eekelen, N., van Veelen, I., van Hees, M., van der Putten, P.: Bursting the bubble (extended abstract). In: MISDOOM 2020. p. 72 (10 2020)
13. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational Linguistics **37**, 267–307 (06 2011)
14. Teernstra, L., van der Putten, P., Noordegraaf-Eelens, L., Verbeek, F.J.: The morality machine: Tracking moral values in tweets. In: Boström, H., Knobbe, A.J., Soares, C., Papapetrou, P. (eds.) Advances in Intelligent Data Analysis XV - 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9897, pp. 26–37 (2016)
15. Wilkinson, J.B., Hausknecht, D.R., Prough, G.E.: Reader categorization of a controversial communication: Advertisement versus editorial. Journal of Public Policy & Marketing **14**(2), 245–254 (1995)
16. Wojdynski, B., Evans, N.: Going native: Effects of disclosure position and language on the recognition and evaluation of online native advertising. Journal of Advertising pp. 1–12 (2016)
17. Zhou, S.: 'Advertorials': A genre-based analysis of an emerging hybridized genre. Discourse & Communication **6**(3), 323–346 (2012)

Accelerating Multi-Agent Learning via Centralized Counting and Efficient Hashing

Jianing Wang, Matthias Müller-Brockhausen, and Aske Plaat

Leiden Institute of Advanced Computer Science, Leiden University, Leiden, the Netherlands
j.wang.35@umail.leidenuniv.nl

Abstract. Exploration is crucial for learning in sparse reward environments such as continuous 2D Navigation or Communicative Navigation. The increased difficulty of multi-over single-agent tasks stems mainly from the increased number of entities requiring coordination and cooperation between each other. To improve cooperation during the exploration phase, we introduce an adaption of the Count-Based method that works centralized, containing all agents' information instead of decentralized. Moreover, we tune a hash function (SimHash) to reduce the high-dimensionality of the continuous navigation environment. With our method, we were able to cut down training time by at least half.

Keywords: Multi-Agent Reinforcement Learning · Exploration · 2D-Navigation

1 Introduction

Learning can be associated with exploration and exploitation. Exploration refers to gaining new information and focusing on long-term gains. Exploitation utilizes current information to maximize short-term benefits. Efficacious exploration is crucial, especially in environments with sparse reward settings. The agents in these environments, with random exploration, barely achieve the tasks and receive learning signals, which is known as the sparse reward problem.

Figure 1 shows two 2D navigation environments with two agents and sparse reward settings. In the 2-Agent Navigation task, agents need to cooperate to cover both landmarks simultaneously. For the Communicative Navigation task, the speaker guides the listener towards the target landmark by uttering a communication signal. In both tasks, agents receive a learning signal only when the landmark is covered.

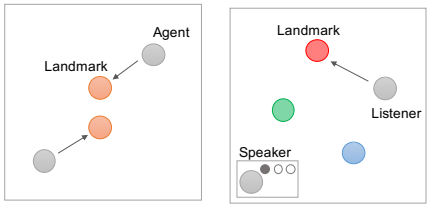


Fig. 1: Multi-agent environments with sparse reward settings: (a) 2-Agent Navigation, (b) Communicative Navigation.

The most common approach to deal with a sparse reward is by introducing an intrinsic reward as a bonus to encourage the agents to explore. For single-agent environments, this has been done extensively [21,26,27]. Since intrinsic reward methods work well in single-agent problems, we are interested in their performance in the multi-agent domain. Multi-agent reinforcement learning (MARL) has intrigued the interest of many researchers in recent years because many real-world applications are naturally modeled as multi-agent learning problems, such as team sport [11], multi-robot control [33], and autonomous vehicles [25]. Contrary to a single agent, cooperation is required to explore an environment efficiently with multiple agents.

To encourage the cooperation between agents, Jaques et al. [8] and Wang et al. [35] strengthen team coordination and communication by encouraging agents to choose actions with more social impact. Compared to their methods that focus on cooperation, we present an exploration method that encourages agents to explore the environment and collaborate with teammates simultaneously. Inspired by the centralized training method [14], we adopt a single-agent intrinsic reward method, Count-Based to Multi-Agent Count-Based (MACB), which considers the information of all the agents for counting. The idea behind the Count-Based method is that agents that can visit more different states in a limited time have a higher chance to find an optimal policy. By counting the occurrences of the joint observations and actions of all agents, the MACB method encourages the agent to visit the states that are new for itself and new for its teammates and therefore achieves simultaneous exploration for both environment and cooperation.

However, all joint observations and actions may only occur once because of the continuous state and action space, making it impossible to determine which state is relatively novel based on counting methods. To solve this problem, we consider using a hash function (SimHash) [32] to map similar state-action pairs to the same hash code before counting.

We evaluate our method with 2-Agent Navigation and Communicative Navigation, which are fully observable and partially observable, respectively. Our results show that the MACB method can help the agents receive the learning signals faster and therefore decrease the number of training episodes that the agents need to master the task by at least half. We also show that our method is easy to implement with existing multi-agent learning algorithms.

2 Related work

For simple RL problems, like MountainCar or CartPole, the basic exploration strategies guarantee finding the optimal decision [12,36]. The ϵ-greedy method [16,34] uses a probability of ϵ to randomly select an action for exploration and a probability of $(1-\epsilon)$ to choose an optimal action. Instead of choosing a random action with a certain probability, the noise-based methods [36] add random noise to action or parameter space directly [4,24]. Random exploration is easy to apply, but it is the least efficient strategy [15,33].

Intrinsic reward strategies are commonly used in hard to explore environments where the agents barely receive learning signals. The intrinsic reward strategies provide bonus rewards as learning signals to the agents through other criteria and therefore boost the progress of learning. Some studies [2,23,28] use the prediction error of different feature spaces to encourage the agents to visit the uncertain parts of the environment. Variational Information Maximizing Exploration (VIME) [6,17] encourages the agents to visit the states, minimizing the uncertainty of the environment dynamics distribution. Count-based methods such as MBIE [29] and MBIE-EB [30] encourage the agent to discover novel states using the state and action count.

Computer scientists took some more RL-related inspiration from the field of psychology [37] by introducing intrinsic rewards in order to encourage cooperative exploration in multi-agent problems. In Jaques et al. [8], the agents are encouraged to select the action which can influence the behavior of the other agents the most. The influence is calculated by how much the selected action can change the distribution over other agents' next actions. In Wang et al. [35], the agent is encouraged to visit the states where that influence the transition distribution of other agents the most. Iqbal et al. [7] use a hierarchical policy where the top-level agent chooses the best among five intrinsic reward functions, and the low-level agents follow this bonus to learn.

In the environments with continuous state and action space, some extensions of the Count-Based method in order to solve the high-dimensional state and action space problem include [1,20], where they propose using a density model to generate pseudo-counts and Tang et al. [32], where they use a hash function to decrease the dimensionality. Instead of the hash function, they also propose a learned hash model (an autoencoder [9,19]) to extract features from the state and reduce the dimensionality. Besides autoencoders, a convolutional neural network that can recognize the pattern of high-resolution images to solve classification tasks can also be used to extract the features [10].

3 Background

We consider an extension of MDP [31] called Markov Games (MGs) [13] to model the MARL problems. For an N agent RL problem, MGs are defined by a set of states S for all agents, sets of actions $A_1, ..., A_N$ and sets of observations $O_1, ..., O_N$ for each agents. The state transition function $P(s_{t+1}|s_t, x_t)$ considers actions from all the agents $x_t = (a_1^t, ..., a_N^t)$ and the state s_t is the concatenation of the observations of all the agents $s_t = (o_1^t, ..., o_N^t)$. We consider the cooperative tasks where all the agents receive the same reward $r_t = R(s_t, x_t)$. Agents aim to maximize the expected reward $R = \sum_{t=0}^{T}(\gamma^t r_t)$, where γ controls the effect that future rewards have on current decisions.

Centralized Critic Algorithm (MADDPG). The centralized critic technique is used to solve the non-stationarity problem in MARL [5,22]. The problem is that all individual policies continuously change during training, making it

impossible to explain the rewards received by each agent with their policies. The multi-agent deep deterministic policy gradient (MADDPG) [14] algorithm utilizes the information from other agents when training the action-value function (centralized critic) but only uses the local information when choosing actions (decentralized actor). Since each agent knows the information of other agents with a centralized critic, it can explain the changes of rewards caused by other agents. Specifically, each agent i has its own centralized action-value function $Q_i(s_t, x_t|\theta^{Q_i})$ which considers the states and actions from all the agents and aims to minimize the loss function:

$$\mathcal{L}(\theta^{Q_i}) = \mathbb{E}_{s_t, x_t, r_t, s_{t+1}}[(Q_i(s_t, x_t|\theta^{Q_i}) - y_t)^2], \tag{1}$$

where

$$y_t = r_t + \gamma \bar{Q}_i(s_{t+1}, \bar{\mu}_1(o_1^{t+1}), ..., \bar{\mu}_N(o_N^{t+1})). \tag{2}$$

The $\bar{Q}_i^{\mu}$ is a copy of Q_i^{μ} that slowly updates towards the critic. Each agent has its own actor $\mu_i(o_i)$, which only considers its local observations o_i. The gradient of the policies is given as:

$$\bigtriangledown_{\theta^{\mu_i}} J(\theta^{\mu_i}) = \mathbb{E}_{s\sim p^{\mu}, x\sim\mu}[\bigtriangledown_{a_i} Q_i(s, a_1, ..., a_i, ..., a_N)|_{a_i=\mu_i(o_i)} \bigtriangledown_{\theta^{\mu_i}} \mu_i(o_i)]. \tag{3}$$

Count-Based Exploration. The MADDPG method adds random action noise to achieve exploration. When an environment has a sparse reward setting, random exploration is the least efficient strategy and may cause the agents to repeatedly explore areas they have been before.

Instead of requiring the agents to complete a task, intrinsic reward methods give a bonus to the agents based on other criteria, such as visiting new states or gathering effective information. When training the policy, a new reward r_t' is used to update the action-value function. It includes an extrinsic reward r_t from the environment and an intrinsic reward r_t^+ [36]:

$$r_t' = r_t + \beta r_t^+ \tag{4}$$

where β is the bonus coefficient that balances exploration and exploitation. The Count-Based exploration strategy uses the state-action count to encourage the agent to visit new state-action. At the time t, the bonus r_t^+ equals the inverse square root count of the state-action pairs:

$$r_t^+(s_t, x_t) = \frac{1}{\sqrt{n(s_t, x_t)}} \tag{5}$$

where $n(s_t, x_t)$ is the number of times this state-action pair has occurred before. With the inverse count bonus, the agent is encouraged to visit the less-visited states. The count is stored in a tabular C.

4 Method

4.1 Multi-Agent Count-Based

The simplest way to adapt Count-Based to the multi-agent domain is assigning each agent i its own count table C_i that uses its local information $n(o_i, a_i)$ for counting. However, the agent may only focus on individual exploration by using the local information and neglect the search for different ways of cooperation with its teammates.

Inspired by the centralized training method, we propose a Multi-Agent Count-Based (MACB) strategy, which takes the joint observations and actions of all the agents $n(o_1, ..., o_N, a_1, ..., a_N)$ for counting and all the agents share a count table C. The joint observations unify all agents information in one central place, allowing cooperation by simultaneously exploring the environment. Sharing a count table can keep the exploration progress of all agents consistent, which helps them achieve the same learning process with the same amount of training.

However, if the environment is with continuous state and action space, all the joint observations and actions will only appear one time. The Count-Based method becomes meaningless if we cannot tell which joint is less visited. This further causes higher storage memory and searching time problems with the count table.

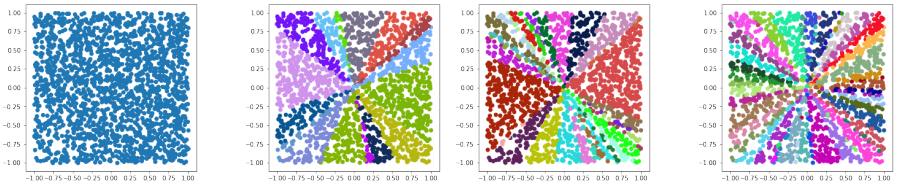
4.2 SimHash Function

Learning from [32], we utilize the SimHash [3] function to discretize a concatenated state-action pair $s||x$ into a k length hash code in the form of $\{-1, 0, 1\}^k$, and use the hash code for counting. The main idea is to map similar state-action pairs into the same hash code. The SimHash function $\phi(s||x)$ discretizes the state-action by the angular distance:

$$\phi(s||x) = \text{sgn}(A \cdot s||x) \in \{-1, 0, 1\}^k, \tag{6}$$

where A is a $k \times D$ matrix with i.i.d. entries sampled from a Gaussian distribution, where D is the size of the state-action $s||x$ and k is the length of the hash code which controls the granularity. To demonstrate how the SimHash function maps similar states into the same code, we randomly draw 2000 points in range $(-1, 1)$ and show the grouping results based on their position with $k = 8, 16, 32$. Figure 2 shows how the SimHash function groups 2-dimensional points angularly. With a larger k, the hash code is longer, and fewer state-action pairs map to the same code. If the hash code is too short, useful information can be lost, which can affect the learning process negatively. Therefore, a suitable k needs to be chosen for optimal results.

After decreasing the scale of state-action pairs using the SimHash function, we can use the corresponding hash code in the MACB strategy. The intrinsic reward



(a) 2000 points (b) $k = 8$ (c) $k = 16$ (d) $k = 32$

Fig. 2: Using a SimHash function to group 2000 points with $k = 8, 16, 32$. Points mapped to the same hash code are grouped in the same color.

is calculated by:

$$r_t^+(s_t, x_t) = \frac{1}{\sqrt{n(\phi(s_t||x_t))}} \tag{7}$$

The pseudo-code of MACB with the SimHash function is shown in Algorithm 1. We update the count of a joint state-action pair in table C after collecting a transition and calculate the new reward after sampling a random transition. The MACB strategy may fail if we update the count after sampling a transition because this transition can be sampled multiple times during training, which will cause the count to increase too quickly, and the intrinsic bonus will vanish after the first few episodes. In addition, the intrinsic reward should not be included in the replay buffer because identical transitions in the replay buffer will have different rewards, leading to inaccuracies as earlier transitions will not have the corresponding rewards for the current situation.

Algorithm 1: Multi-Agent Count-based (MACB)

Initialize multi-agent learning algorithm (e.g. MADDPG)
Initialize an empty hash tables C where the new key initialize with value 0
Initialize hyper-parameters β for trade-off and k for hash code granularity
Initialize matrix $A \in \mathbb{R}^{k \times D}$ with i.i.d. entries sample from a normal distribution
for *episode* $= 1$ *to* M **do**
  for $t = 0$ *to* T **do**
    Collect transition (s_t, x_t, r_t, s_{t+1}) and store in the replay buffers
    Compute hash code using SimHash function
     $\phi(s_t||x_t) = \text{sgn}(A \cdot s_t||x_t) \in \{-1, 1\}^k$
    Update the count in the table C, $n(\phi(s_t||x_t)) = n(\phi(s_t||x_t)) + 1$
    for *agent* $i = 1$ *to* N **do**
      Sample a minibatch of transitions (s_j, x_j, r_j, s_{j+1}) from replay buffers
      Compute hash code of each state-action pair $\phi(s_j||x_j)$
      Calculate the new reward $r'_j = r_j + \beta r_j^+$ where $r_j^+ = \frac{1}{\sqrt{n(\phi(s_j||x_j))}}$
      Update critic and actor using the new reward r'_j
    end
  end
end

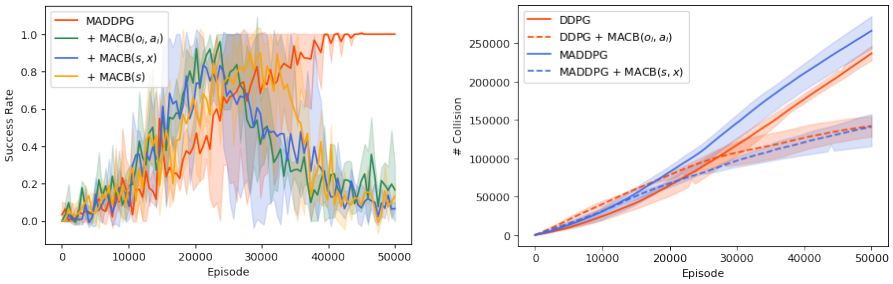
5 Experiments

We evaluate MACB in 2 different cooperative multi-agent tasks, 2-Agent Navigation and Communicative Navigation [18], as shown in Figure 1. Both of the environments are 2-dimensional with a continuous state and action space. 2-Agent Navigation is fully observable, which means that the agent can see all relevant information that it needs to make a decision. Communicative Navigation is a partially observed environment, where only the speaker knows which landmark is the target, and the listener has to decipher it based on the speaker's signal.

In both tasks, we set 20 timesteps for each episode. After an episode of training, we run ten more episodes without random action noise for evaluation. All the results are smoothed and averaged over three random seeds with a 75% confidence interval. Our code can be found in Github [1] and we include the hyper-parameters for the learning algorithm in Appendix A.

5.1 Performance in the fully observed environment

Figure 3a shows the average success rate of MADDPG with and without MACB on the partially observed 2-Agent Navigation task. Without the help of MACB, MADDPG learns gradually over episodes and fully grasps the problem at around 4×10^4 episodes. All 3 MACB variants accelerate learning and reach success rate convergence earlier (2 at 2×10^4 and 1 at 3×10^4 episodes). However, if we continue training after convergence, the success rate gradually decreases, which can be addressed by using Early Stopping.



(a) Success rate (b) Number of collisions

Fig. 3: (a) The success rate in the 2-Agent Navigation problem with and without MACB exploration. (b) The number of collisions of DDPG and MADDPG with and without the MACB method. The MACB method can promote the learning process and decrease the number of collisions.

Since the environment is already fully observable, the performance of local [MACB(o_i, a_i)] vs global [MACB(s, x)] information does not differ much. Moreover, contrary to [32], state-action pair counting improves performance in our

[1] https://github.com/JianingWang99/CentralizedCountBased

experiments. The probable reason for this difference is that [32] applied it in a single-agent environment.

We compare the accumulated number of collisions between DDPG and MADDPG with and without MACB in Figure 3b. According to [14], the MADDPG agents only have half of the number of collisions than DDPG agents, but our result shows that the number of collisions of the MADDPG is more than that of DDPG. However, after applying the MACB strategies, the accumulated number of collisions vastly decreases. These results indicate that the MACB exploration can help the agents find the optimal cooperation strategies within fewer episodes, and in turn, the total number of collisions decreases.

5.2 Performance in the partially observed environment

Figure 4 shows the success rate of MACB strategies in the partially observed environment with 2 different reward settings. With sparse rewards, the MADDPG agents learn slowly and only surpasses a 40% success rate after 4×10^4 episodes of training. While with the help of MACB(o_i, a_i) the success rate increases to around 70% at 4×10^4 episodes, the agents with MACB(s, x) can reach 100% success rate with only 1.5×10^4 episodes. Both MACB strategies have the same β (0.8) and k (512). This underlines the positive effect of centralization in partially observed environments.

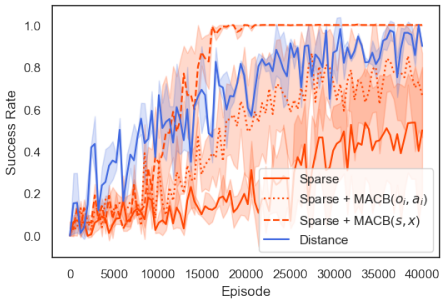


Fig. 4: The success rate of MADDPG and MACB strategies in the Communicative Navigation with sparse and distance reward settings.

In the sparse reward setting, the agent requires more time before receiving a steady learning signal. Distance rewards seem to mitigate this and enabling instantaneous learning. We can see that the MACB method can reduce the number of episodes that the agents require to receive learning signals. And once the agent starts learning, the success rate increases faster than the agent in the dense reward environment.

5.3 Trade-off between exploration and exploitation

Table 1 concludes the average success rate and the count-1 percentage (sr, c-1) after 4×10^4 episodes of training with different combinations of β and k. The Count-1 percentage reflects how many state-action pairs only appear one time in the count table. We evaluate using the Communicative Navigation environment.

Without the help of MACB, the MADDPG agents reach a 47% success rate. With the help of MACB, most of the time, the success rate is higher than 47%

with different hyper-parameters. When k is 32, the success rate just increases a little with different β values. The results indicate that a hash code will lose important and relevant information if k is too small, making it slower to learn the task. We also see that the count-1 percentage increases as k , and when k is 512, the count-1 percentage converges to 100%, and the success rates surpass 75% in most of the cases.

When tuning k, we can increase the value of k until its count-1 percentage converges at 100%. When k is too big, the hash function may lose its meaning and the search time and storage memory becomes high. With a larger value of k, the agents need to explore more state-action pairs and require a larger β. In addition, we control β smaller than 1 because we do not want the exploration bonus to overwhelm the extrinsic reward, as it would lead to the agents only exploring and never exploiting.

Table 1: The table concludes the success rate (sr) and count-1 percentage (c-1) of the MACB with MADDPG in the Communicative Navigation with different ratios of exploration (β) and length of hash code (k).

β	**0.0**	**0.05**	**0.2**	**0.4**	**0.8**
k	sr, c-1	sr, c-1	sr, c-1	sr, c-1	sr, c-1
-	47% , -	-	-	-	-
32	-	59%, 67%	60%, 68%	56%, 71%	53%, 74%
64	-	**72%**, 91%	60%, 91%	34%,91%	**72%**, 94%
256	-	25%, 99%	**82%**, 99%	43%,99%	**81%**, 99%
512	-	**75%**,100%	48%,100%	**75%**,100%	**100%**,100%

6 Conclusion

Our work has succeeded in improving the exploration in multi-agent environments with a sparse reward setting, specifically: 2-Agent Navigation and Communicative Navigation (see Figure 1). By centralizing the count table of our Count-Based method, we have improved cooperation between agents. Moreover, we have successfully reduced the high-dimensionality of the continuous environment without losing training-relevant information by applying SimHash. After tuning its two parameters β and k, we were able to accelerate learning dramatically. For future work, there are two interesting paths to pursue. First, it remains to be seen what other multi-agent tasks our extensions work well on and how they perform in settings with more than just two agents. Second, solving the same task solely from an image input. SimHash requires knowledge about the entities' position, so an auto-encoder extracting them would be required. Alternatively, one could utilize a density model to predict the pseudo-counts of state-action pairs instead of using a table to record the counts directly.

References

1. Bellemare, M.G., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., Munos, R.: Unifying count-based exploration and intrinsic motivation. arXiv preprint arXiv:1606.01868 (2016)
2. Burda, Y., Edwards, H., Storkey, A., Klimov, O.: Exploration by random network distillation. arXiv preprint arXiv:1810.12894 (2018)
3. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing. pp. 380–388 (2002)
4. Fortunato, M., Azar, M.G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., et al.: Noisy networks for exploration. arXiv preprint arXiv:1706.10295 (2017)
5. Gronauer, S., Diepold, K.: Multi-agent deep reinforcement learning: a survey. Artificial Intelligence Review pp. 1–49 (2021)
6. Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., Abbeel, P.: Vime: Variational information maximizing exploration. arXiv preprint arXiv:1605.09674 (2016)
7. Iqbal, S., Sha, F.: Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning. arXiv preprint arXiv:1905.12127 (2019)
8. Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P., Strouse, D., Leibo, J.Z., De Freitas, N.: Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In: International Conference on Machine Learning. pp. 3040–3049. PMLR (2019)
9. Krizhevsky, A., Hinton, G.E.: Using very deep autoencoders for content-based image retrieval. In: ESANN. vol. 1, p. 2. Citeseer (2011)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)
11. Kurach, K., Raichuk, A., Stańczyk, P., Zajac, M., Bachem, O., Espeholt, L., Riquelme, C., Vincent, D., Michalski, M., Bousquet, O., et al.: Google research football: A novel reinforcement learning environment. arXiv preprint arXiv:1907.11180 (2019)
12. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)
13. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Machine learning proceedings 1994, pp. 157–163. Elsevier (1994)
14. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. arXiv preprint arXiv:1706.02275 (2017)
15. McFarlane, R.: A survey of exploration strategies in reinforcement learning. McGill University (2018)
16. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. pp. 1928–1937. PMLR (2016)
17. Mohamed, S., Rezende, D.J.: Variational information maximisation for intrinsically motivated reinforcement learning. arXiv preprint arXiv:1509.08731 (2015)
18. Mordatch, I., Abbeel, P.: Emergence of grounded compositional language in multi-agent populations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)

19. Ng, A., et al.: Sparse autoencoder. CS294A Lecture notes **72**(2011), 1–19 (2011)
20. Ostrovski, G., Bellemare, M.G., Oord, A., Munos, R.: Count-based exploration with neural density models. In: International conference on machine learning. pp. 2721–2730. PMLR (2017)
21. Oudeyer, P.Y., Kaplan, F.: What is intrinsic motivation? a typology of computational approaches. Frontiers in neurorobotics **1**, 6 (2009)
22. Papoudakis, G., Christianos, F., Rahman, A., Albrecht, S.V.: Dealing with non-stationarity in multi-agent deep reinforcement learning. arXiv preprint arXiv:1906.04737 (2019)
23. Pathak, D., Agrawal, P., Efros, A.A., Darrell, T.: Curiosity-driven exploration by self-supervised prediction. In: International Conference on Machine Learning. pp. 2778–2787. PMLR (2017)
24. Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R.Y., Chen, X., Asfour, T., Abbeel, P., Andrychowicz, M.: Parameter space noise for exploration. arXiv preprint arXiv:1706.01905 (2017)
25. Shalev-Shwartz, S., Shammah, S., Shashua, A.: Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295 (2016)
26. Singh, S., Barto, A.G., Chentanez, N.: Intrinsically motivated reinforcement learning. Tech. rep., MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE (2005)
27. Singh, S., Lewis, R.L., Barto, A.G., Sorg, J.: Intrinsically motivated reinforcement learning: An evolutionary perspective. IEEE Transactions on Autonomous Mental Development **2**(2), 70–82 (2010)
28. Stadie, B.C., Levine, S., Abbeel, P.: Incentivizing exploration in reinforcement learning with deep predictive models. arXiv preprint arXiv:1507.00814 (2015)
29. Strehl, A.L., Littman, M.L.: A theoretical analysis of model-based interval estimation. In: Proceedings of the 22nd international conference on Machine learning. pp. 856–863 (2005)
30. Strehl, A.L., Littman, M.L.: An analysis of model-based interval estimation for markov decision processes. Journal of Computer and System Sciences **74**(8), 1309–1331 (2008)
31. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
32. Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, X., Duan, Y., Schulman, J., De Turck, F., Abbeel, P.: # exploration: A study of count-based exploration for deep reinforcement learning. In: 31st Conference on Neural Information Processing Systems (NIPS). vol. 30, pp. 1–18 (2017)
33. Thrun, S.B.: Efficient exploration in reinforcement learning (1992)
34. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 30 (2016)
35. Wang, T., Wang, J., Wu, Y., Zhang, C.: Influence-based multi-agent exploration. arXiv preprint arXiv:1910.05512 (2019)
36. Weng, L.: Exploration strategies in deep reinforcement learning. lilianweng.github.io/lil-log (2020), https://lilianweng.github.io/lil-log/2020/06/07/exploration-strategies-in-deep-reinforcement-learning.html
37. Wiersma, U.J.: The effects of extrinsic rewards in intrinsic motivation: A meta-analysis. Journal of occupational and organizational psychology **65**(2), 101–114 (1992)

A Appendix

A.1 Training Details

The networks for actors and critics are with two hidden layers with the size of 400 and 300. The activation function for both actor and critic is ReLU. After collecting 100 transitions, we begin updating the network parameters at each time step. An episode consists of 20 time steps. After an episode of training, we evaluate the algorithms with 10 more episodes without exploration action noises. All the results are averaged over 3 random seeds. The hyper-parameters used in the experiments are summarized in Table 2.

Table 2: Hyper-parameters used in experiments

Hyper-parameter	**Value**
Buffer size	10^6
Batch size	100
Time-step per Episode	20
Learning rate for optimizer	0.001
γ	0.99
τ	0.005

Regular Decision Processes for Grid Worlds

Nicky Lenaers and Martijn van Otterlo

Open University, The Netherlands

Abstract. Markov decision processes are typically used for sequential decision making under uncertainty. For many aspects however, ranging from *constrained* or *safe* specifications to various kinds of temporal (*non-Markovian*) dependencies in task and reward structures, extensions are needed. To that end, in recent years interest has grown into combinations of reinforcement learning and temporal logic, that is, combinations of flexible behavior learning methods with robust verification and guarantees. In this paper we describe an experimental investigation of the recently introduced *regular decision processes* that support both non-Markovian reward functions as well as transition functions. In particular, we provide a tool chain for regular decision processes, algorithmic extensions relating to online, incremental learning, an empirical evaluation of model-free and model-based solution algorithms, and applications in regular, but non-Markovian, grid worlds.

Keywords: sequential decisions · safe reinforcement learning · non-Markovian dynamics · regular decision process · linear temporal logic

1 Introduction

Sequential decision making under uncertainty, often simply denoted by its core algorithmic subfield *reinforcement learning* (RL) [36, 39, 34], has been showing a huge amount of progress the last decades. Among the recent breakthroughs is the progression of DeepMind's RL methods solving the board game Go [32], chess, Atari computer games, the real-time strategy game StarCraft II, and lately chip design [26]. The algorithms employ combinations of (Monte Carlo) planning and value function approximation using deep neural networks.

Underlying typical RL systems is the *Markov decision process* (MDP) [30] in which *states* carry all necessary information to choose (optimal) *actions*. The *Markov property* dictates that given the present, the future is *independent* of the past. To scale to more complex problems, one can exploit *structure* in the space of state(-action) spaces, or policies or value functions, to utilize abstractions and approximations, for example as *value function approximation*, state space abstractions [37], and hierarchical decompositions, cf. [36]. Many current *deep* RL algorithms too assume the environment behaves as an MDP [38].

To scale to larger problems, the Markov property is no longer adequate, and one may require dependence on a *history* of events and observations. For example, consider a robotic waiter working in a restaurant. It needs to deliver food and

beverages to tables, but only *after* it has been requested by guests, and at the end the guests need to pay the price of the items delivered earlier. However, keeping a history of every possible event that ever occurred soon becomes practically infeasible. One well-known class of non-Markov MDP extensions is the *partially observable* MDP [33] in which the current state can be represented as a *probability distribution* over (latent) state features, denoted a *belief state*. Despite the existence of effective POMDP algorithms, many in robotics domains, the general class of POMDPs is computationally much more complex than MDPs, it is not easy to decide what the belief state should include exactly, and how much history should be included, and updating and interpreting the belief states is non-trivial.

A prominent RL direction [23] is to model dependence on the arbitrary past *explicitly* resulting in non-Markovian variants of MDPs. Inspired by seminal work [3] the idea is to utilize modern logical languages such as *linear temporal logic* [29] to represent goals and reward functions over past traces, and to employ formal computer science techniques (e.g. *automata*, *verification* and *model-checking*) in decision making. A core idea here is to *compile* a temporal specification of a reward function into an automaton that *monitors* the fulfillment of the temporal formula. Monitors allow for *compiling* the original non-Markov problem back into the MDP framework such that all existing algorithms, including deep RL, can be employed. This fruitful marriage of RL and formal verification combines flexible behavior learning algorithms with formal performance guarantees.

One motivation for employing temporal logic in RL comes from the ability to elegantly specify complex reward structures as in the waiter example, where earnings depend on an ordered series of events in the history. Another, more general, motivation is the need to *constrain* RL behaviors using (declarative) knowledge about which behaviors are desired or considered *safe* [15], for example to teach an autonomous car how to drive while still obeying traffic rules. Transparent safety of learned behaviors is often part of a general desire for AI systems to behave *responsibly* and *explainable* [28, 24, 19].

In this paper we empirically investigate algorithmic variations in one of the most recently introduced models, *regular decision processes* (RDP) [6], in which reward functions *and* transition functions can be specified using temporal logic. We employ RDPs specifically for *grid worlds*, which are archetypical problem scenarios in RL and allow for focused experimentation with new representations and algorithms. More specifically, our contributions are i) a novel *tool chain* implementing RDPs, utilizing exisiting algorithms and tools for RL and model checking, ii) an empirical investigation of the recently introduced RDPs in grid worlds, iii) algorithmic RL extensions to learn RDP behaviors based on Monte Carlo value estimation and incremental (online) compilation of RDPs, and iv) initial steps towards an (empirical) investigation of the trade-offs between temporal logical specifications and the complexity of learning. The paper is organized as follows: we first provide all necessary background in the next section, after which we discuss our approach in Section 3, then we continue with an extensive experimental evaluation in Section 4 and we conclude in Section 5.

2 Background

Here we will formalize MDPs and basic solution algorithms, after which we introduce non-Markov reward functions and their corresponding temporal logic formalizations. Furthermore we introduce the general compilation of logical specifications into automata functioning as monitors that can be combined with the original MDP into *extended* MDPs, which can be solved using off-the-shelf solution methods. In addition, we describe automata-based *shaping* techniques to deal with the resulting sparse MDPs. Last we introduce RDPs, which support non-Markovian aspects in both reward and transition functions.

2.1 Markov Decision Processes

An MDP M is a tuple $M = \langle S, A, T, R \rangle$, where S is the set of states, A the set of actions, $T : S \times A \times S \to [0, 1]$ the *transition function* yielding a transition probability and $R : S \times A \times S \to \mathbb{R}$ the real-valued *reward function*. Actions only applicable in state s are denoted $a \in A(S)$. A *policy* maps to each state $s \in S$ an action $a \in A$ and is denoted π. Additionally, a *discount factor* $\gamma \in [0, 1]$ is used to discount rewards obtained in the future.

As said, MDPs adhere to the *Markov Property*: given the present (s_t), the future (s_{t+1}) is independent on the past (s_{t-1}). In other words, everything that is needed to learn from the past is *embedded* in the present state s_t. The Markov Property holds for all states $s \in S$ and is formally expressed as:

$$p(s_{t+1}|s_t) = p(s_{t+1}|s_1, s_2, \ldots, s_t)$$

A labelling function $\mathcal{L} : S \to 2^P$, where P is a finite set of atomic propositions and S the set of states enables a state representation using *features*.

Solving an MDP comprises computing an optimal policy. A policy is optimal iff it maximizes the expected discounted sum of rewards for every state $s \in S$. Methods for solving decision making problems are generally divided into *model-based* and *model-free* methods [34]. Model-based methods, generally called *dynamic programming* (DP), can employ the full model (T and R) to *plan* optimal sequences of actions. Model-free methods, generally called *reinforcement learning* (RL), do not have knowledge of the model and require sampling, i.e. trial-and-error learning and use that experience to find optimal policies.

Dynamic Programming (DP) methods such as *value and policy iteration* find optimal policies typically by employing a *value function* that expresses for each state *how good* is it for the agent to be in that particular state, and it represents the (expected) discounted future reward that can be obtained from that state, by employing a particular policy. The equation used to calculate a state value is known as the *Bellman Equation*, which formalizes how a state's value, denoted $v(s)$, is evaluated in terms of expected returns, expressing a relationship between the value of a state and the values of its successor states. DP algorithms use it *iteratively* to update the value of all states until convergence to the *optimal value*

function $v^*(s)$ using the following *Bellman Optimality Equation*:

$$v^*(s) = \max_{a \in A} \sum_{s' \in S} T(s'|s,a) \left[R(s,a,s') + \gamma \, v^*(s') \right]$$

An optimal action a for s is computed using $v^*(s)$, T and R.

Where DP methods are concerned with *computing* a value function, RL tries to *learn* value functions using returns obtained from *interaction* with the MDP. In order to find a policy in absence of a model, one needs the *state-action value* for each action $a \in A$ in state $s \in S$, denoted $q(s,a)$, in order to determine the best policy. A straightforward extension of the previous update rule results in $q^*(s,a) = \sum_{s' \in S} T(s'|s,a) \left[R(s,a,s') + \gamma \max_{a' \in A} q^*(s',a') \right]$. *One-step* RL algorithms employ it to *update* action-values after each step in the environment and select their actions based on $\pi^*(s) = \arg\max_a q(s,a)$.

In addition to *bootstrapping* methods above, where values of states (and actions) are computed using other values, one can employ more unbiased estimation methods for model-free RL such as *Monte Carlo estimation* (MC) in which a value is estimated based on the average return of full sample traces in the MDP, cf. [34]. In Section 3 we employ MC as our model-free RL algorithm for RDPs.

2.2 Non-Markovian Decision Processes

If rewards depend on more than just the current state, we end up with Non-Markovian Reward Decision Processes (NMRDPs) [3], a subset of Non-Markovian Decision Processes (NMDPs). Temporal logic can be used to specify the conditions under which reward is obtained. As with MDPs, the states of an NMRDP can be enhanced by labelling function $\mathcal{L} : S \to 2^P$ and propositions P, where each state $s \in S$ is a valuation over P, thus $s \in 2^P$.

Formally, an NMRDP is denoted as the tuple $M = \langle S, A, T, \bar{R} \rangle$, where S, A and T are as in an MDP, and $\bar{R}$ is defined as $\bar{R} : (S \times A)^* \to \mathbb{R}$. In words, the reward is specified as a real-valued function over finite state-action sequences, or *traces*, where a trace captures the history of states and is denoted $h = \langle s_0, \ldots, s_k \rangle$. Because the reward is now dependent on the full history, it no longer fits to define state or state-action values as before. Instead, a temporally extended reward function for a given trace h and reward formulae φ is [4]:

$$\bar{R}(h) = \sum_{1 \leq i \leq n \,:\, h \models \varphi_i} r_i \tag{1}$$

where the set of pairs $\{(\varphi_i, r_i)_{i=1}^n\}$ is assumed to be specified for $\bar{R}$. That is, an agent receives reward r_i at state $s \in S$ of trace h that satisfies temporal formula φ_i. The value of a trace h is in turn defined as the accumulation of rewards obtained during trace traversal, possibly discounted by discount factor γ [4]. The value of such a trace can now formally be defined as follows:

$$v(h) = \sum_{k=1}^{|h|} \gamma^{k-1} \, \bar{R}(\langle h(1), h(2), \ldots, h(k) \rangle)$$

where discount factor $\gamma \in [0, 1]$ as usual and $h(k)$ denotes the pair (s_{k-1}, a_k). Because NMRDPs define the value of traces instead of individual states, a policy no longer maps states to actions as before. Instead, a policy for an NMRDP is a mapping from histories to actions. The value of a policy in terms of expected return thus becomes the expected discounted sum of rewards over a possibly infinite amount of traces. The distribution over traces is defined by the initial state s_0, the transition function T and policy π. The expected value of infinite traces can formally be defined as $v_\pi(s) = E_{h \sim M,\pi,s_0} v(h)$.

2.3 Temporal Logic, Automata and Product MDPs

Temporal logic to express non-Markovian aspects has a history [3, 29] containing, e.g., Linear Temporal Logic [29] (LTL). It uses the standard Boolean connectives of propositional logic, i.e. $\wedge$, $\vee$ and $\neg$, with the addition of temporal connectives G (*always*), F (*eventually*), X (*next*) and U (*until*). More recent variations restrict to finite traces: *Linear Temporal Logic over Finite Traces*, denoted LTL_f, and *Linear Dynamic Logic over Finite Traces* which allows for regular expressivity [12]. Using LDL_f, goals can be as expressive as regular expressions while at the same time providing a more attractive specification syntax. Formally, LDL_f formulae ϕ can be built using an atomic property tt for the logical *true*, a propositional formula φ and a path expression ρ, which is a regular expression over propositional formulae ϕ. In addition to regular expression constructs, ρ uses a test construct $\varphi?$, indicating to only continue evaluation when φ evaluates to *true*. The LDL_f formalism, as presented by [12], is expressed in Equations (2) and (3).

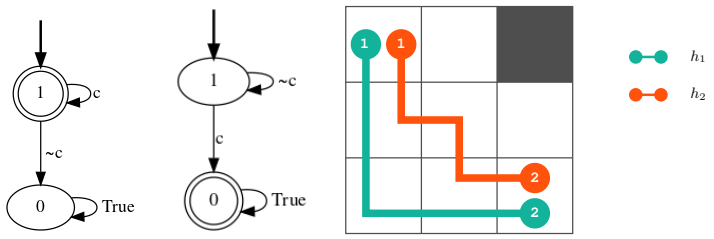
$$\varphi ::= tt \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle \varrho \rangle \varphi \tag{2}$$

$$\varrho ::= \phi \mid \varphi? \mid \varrho_1 + \varrho_2 \mid \varrho_1; \varrho_2 \mid \varrho^* \tag{3}$$

Intuitively, one may interpret LDL_f formula $\langle \varrho \rangle \varphi$ as stating that, from the current step in the trace, there exists *at least one* (cf. $\exists$) execution path that satisfies regular expression ϱ such that the last step in the trace satisfies φ. Conversely, $[\varrho]\varphi$ states that, from the current step in the trace, *all* (cf. $\forall$) execution paths satisfying regular expression ϱ are such that the last step in that execution path satisfies φ. For example, to formalize the property of a robotic waiter to always serve guests after they have placed an order, the formula $[true*](order \rightarrow \langle true^*; served \rangle) end$ can be used.

Temporal formulae specified using LDL_f can be compiled into Deterministic Finite Automata (DFA) [4]. Formally, a DFA for formula φ is denoted $A_\varphi = \langle 2^P, Q, \delta, F, q_0 \rangle$, where 2^P is the input alphabet containing all truth assignments to propositions in P, Q is the state space, δ the transition function, F the set of accepting states and q_0 the initial state.

Core properties that can be expressed in LDL_f are *safety* and *liveness* [12]. A safety property is used to indicate that *something bad should never happen*, or *something good always holds*, and can be expressed as $[true^*]\langle c^* \rangle end$, where c indicates the good condition and the asterisk (*) indicates c holds at every step



(a) Safety (b) Liveness (c) Grid World

Fig. 1: (left) Automata for LDL_f formulae: a) $[true^*]\langle c^*\rangle end$, b) $\langle true^*; c; true^*\rangle end$ (right) A grid world modeled as an RDP

up until and including the last step of the trace. That is, *until the end of the trace*, c holds. Conversely, a liveness property indicates that some condition should be met *before the end of the trace* and can be expressed as $\langle true^*; c; true^*\rangle end$, where c is the condition to be met. In words, *eventually before the end of the trace*, c holds. Figures 1a and 1b visualize this.
Solving an NMRDP $M = \langle S, A, T, \{(\varphi_i, r_i)_{i=1}^m\}\rangle$, with temporal formulae φ_i and r_i the corresponding rewards, is tackled by formulating the *extended* MDP M' as $M' = \langle S', A, T', R'\rangle$ that is *equivalent* to M in the sense that states can be mapped in such a way that the mapping yields identical transition probabilities for T and T'. Each formula φ_i is compiled into an equivalent automaton, as in Figures 1a and 1b, and the *cross-product* between the original NMRDP M and these automata is computed, resulting in the extended MDP M'. Some straightforward choices should still be made about discounting to prevent infinite reward exploitation and whether rewards belonging to a formula φ can be obtained only once or multiple times. We omit formal details of this standard construction (but cf. [4, 21]) and refer here to an example later in this paper: Figure 5 shows a grid world MDP where a red square needs to be avoided, something which is specified using the LDL_f formula $\varphi \equiv [true^*]\langle(\neg x_{is1} \wedge \neg y_{is2})^*\rangle end$, and where the extended MDP depicted in Figure 9 is the result of the cross product between the automaton representing φ, the grid world MDP, and the automaton representing an additional formula expressing a reward of $+50$ when reaching the top right corner. Note that the extended model is again an MDP where typical RL and DP algorithms can be employed.

2.4 Regular Decision Processes: Non-Markovian Dynamics

The concept of an NMRDP can be extended to a decision process in which not only the reward function, but the transition function too can depend on the arbitrary past, and where both are represented using a logic like LDL_f As described in Section 2.3, these, in turn, can be compiled into automata, which

allows for rewards and transitions to be *monitored*, and compiled into product models yielding an MDP. Such non-Markovian transitions were introduced in regular decision processes (RDP) [6], which is a fully observable, probabilistic, non-Markovian, sequential decision making model, where successor states and rewards can be stochastic functions of the entire history. Just like before, RDP states are endowed with a labeling function over a set of predicates.

An RDP M is defined as the tuple $M = \langle P, S, A, Tr_L, R_L, s_0 \rangle$, where P is the set of propositions that induces state-space S with initial state s_0, A the set of actions, Tr_L the transition function and R_L the reward function, where both Tr_L and R_L are now non-Markovian. Transition function Tr_L is defined by a finite set T of quadruples of the form $(\varphi, a, P', \pi(P'))$, where φ is an LDL_f formula over P, $a \in A$ an action, $P' \subseteq P$ the set of propositions $p \in P$ that are affected by a when φ holds and $\pi(P')$ the distribution over proposition in P' that describe the post-action distribution. The reward function R_L is specified using a finite set R of pairs (φ, r), where φ is an LDL_f formula over propositions in P and $r \in \mathbb{R}$ a real-valued reward. It is assumed that for the quadruples in T, the value of variables not in P' are not affected by action a [6]. If the set $\{(\varphi_i, a, P'_i, \pi_i(P')) \mid i \in I_a\}$ defines all quadruples for a, then all formulae φ_i must be *mutually exclusive* such that $\varphi_i \wedge \varphi_j$ is inconsistent for $i \neq j$. In other words, no two formulae φ_i and φ_j can hold at once if both apply to action a and φ_i and φ_j are not identical. In addition, let $s'|_{P'}$ denote the restriction of s' to properties in P'. Then, Tr_L is defined as $Tr_L((s_0, \ldots, s_k), a, s') = \pi(s'|_{P'})$ if quadruple $(\varphi, a, P', \pi(P'))$ exists such that $s_0, \ldots, s_k \models \varphi$ and s_k and s' agree on all variables in $P \setminus P'$. That is, given trace $s_0, \ldots, s_k$, action a and quadruple $(\varphi, a, P', \pi(P'))$ with formula φ that is satisfied by $s_0, \ldots, s_k$, s' is a possible next state if it assigns the same value to all propositions not in P'. If this is the case, then the transition probability equals the probability π assigns to $s'|_{P'}$. In all other cases, $Tr_L((s_0, \ldots, s_k), a, s') = 0$.

As an illustration, consider Figure 1c, outlining a 3×3 grid world with the upper-left state s_{11} being the initial state and the upper-right state s_{31} being a terminal state. Let us define a transition that intuitively states that, when an agent goes east in the bottom-left state s_{13} and ends up in the bottom-center state s_{23}, immediately followed by going east *again* in s_{23}, the probability of ending up in the bottom-right state s_{33} is set to 0.1, denoted $\pi(s_{33}|\{x_{is2}, x_{is3}\}) = 0.1$. Otherwise, $Tr(s_{23}, e, s_{33}) = 1$. In other words, the transition from s_{23} to s_{33} depends on the transition from s_{13} to s_{23}. In addition, the propositions affected by this transition are defined by P' such that $P' \subseteq P = \{x_{is2}, x_{is3}\}$. All other propositions are not affected by said transition. Both transitions can be captured by LDL_f formula φ_1 and φ_2 as $\varphi_1 = \langle true^*; \neg x_{is1} \vee \neg y_{is3}; x_{is2} \wedge y_{is3} \rangle end$ and $\varphi_2 = \langle true^*; x_{is1} \wedge y_{is3}; x_{is2} \wedge y_{is3} \rangle end$. Given φ_1 and φ_2, we can define a quadruple for e that uses φ_1 or φ_2 respectively as $(\varphi_1, \{x_{is3} \wedge y_{is3}\}, 1)$ and $(\varphi_2, e, \{x_{is3} \wedge y_{is3}\}, 0.1)$. For brevity, we assume these are the only quadruples for e, conforming to exhaustiveness and mutual exclusion [6]. Then, let us define two traces h_1 and h_2 that each reach s_{33} differently as $h_1 = \langle s_{11}, s_{12}, s_{13}, s_{23}, s_{33} \rangle$ and $h_2 = \langle s_{11}, s_{12}, s_{22}, s_{23}, s_{33} \rangle$. Then, using the

aforementioned quadruple for e, the affected propositions P' and the definition of Tr_L, i.e. $Tr_L((s_0, \ldots, s_k), a, s') = \pi(s'|_{P'})$, the transition functions on h_1 and h_2 from s_{23} to s_{33} can be calculated as $Tr_L(h_1, e, s_{33}) = \pi(s_{33}|\{x_{is2}, x_{is3}\}) = 1$ and $Tr_L(h_2, e, s_{33}) = \pi(s_{33}|\{x_{is2}, x_{is3}\}) = 0.1$.

Solving an RDP involves the well known construction of an *extended* MDP as a product of all automata monitoring the satisfaction of (transition and reward) LDL_f formulae combined with the initial RDP state space [4, 12], resulting in an MDP that can again be solved by off-the-shelf algorithms. Note that, because of the combinatorial nature of this construction, the extended MDP does not necessarily scale well. The equivalence between the RDP and the constructed MDP entails that optimal policies found in the constructed MDP can be mapped back to the RDP, thus yielding optimal policies for the initial RDP.

The product models employed in non-Markovian decision process solutions grow quickly with the number of formulas, see the example in Section 2.3. The result of non-Markovian dependencies is that paths to receiving rewards can become long, and complicate typical bootstrapping RL methods and exploration. One general solution for MDPs is *reward shaping* [27] (RS): giving intermediate rewards to speed up learning, with the restriction that the extra rewards do not alter the optimal policy. So-called *potential-based* RS replaces the original reward function $R : S \times A \times S \to \mathbb{R}$ by an alternative reward function $R'(s, a, s') + F(s, a, s') \to \mathbb{R}$, where $F(s, a, s')$ is a *shaping reward function*. In turn, this function can be applied to *potential-based* RS of the form $F(s, a, s') \to \gamma\Phi(s') - \Phi(s)$ for some $\Phi : S \to \mathbb{R}$. The way in which RS is applied inherently depends on the representation of the reward function. For NMRDPs an opportunity arises to utilize the structure of the DFA representing a reward function [10]. Every step in the extended MDP can be given a reward proportional to the *distance* in that DFA to an accepting state (i.e. when the original reward would be given).

2.5 Related Work

The typical MDP context is well studied and there is an abundance of algorithms and representations [36, 39, 30, 34]. Endowing MDPs with non-Markovian goal and reward functions has a history with seminal work on model-based settings [3, 35] and more recently several subclasses are considered (e.g. probabilistic vs. deterministic) [4, 5]. The most recent addition to the field are the general *regular decision processes* [6] we employ here. One aim of all these methods is to scale MDPs to more complex problems. However, another main reason to utilize temporal logics for reward specifications is that it opens up many new possibilities for *reward function engineering*, resulting in more intuitive and technically useful ways to specify tasks and goals. A more general view, based on automata as *transducers* [9] improves on the technical part by merging the non-Markovian parts into a single structure.

The use of temporal logics [29, 12] in model-free RL settings is a recent trend [23], and comes with additional requirements since the model of the environment is unknown. Many ideas here come from *constrained* or *safe* [15] forms of RL, where the policy space is restricted either before learning, or during action

selection, based on a notion what are (un)desired actions. Safety issues have obvious connections with *model checking* [16] and some recent RL approaches instantiate that connection for safe RL [2, 13]. Very recently, several approaches have appeared combining formal temporal logic with RL [22, 4, 10, 7, 11, 9]. Some focus more on the representational devices such as *reward machines* [7], some study additional mechanisms such as *shaping* to aid in the more complex learning process [27], and others introduce variants such as *geometric* LTL to capture a different semantics of goals [25]. Overall, variations exist in different logics, different underlying automata (e.g. DFA vs Mealy) and inference algorithms, and different RL algorithms to solve the resulting extended MDPs.

The meaning of "model-free" has variations here, since one can can assume that nothing is known, or that at least the reward formula is (which is quite believable when we want an agent to adhere to certain rules or restrictions). In the latter case one can use the monitor automata states as extra state information and apply any form of deep function approximation [18]. In general, reward and transition functions may need to be learned from traces for fully general RL systems. In the temporal logic settings we describe, this typically amounts to *automaton induction* algorithms that can work on examples of traces (positive or negative) in deterministic or even probabilistic settings, which contains notoriously hard settings, but some promising work is emerging [8, 20, 14]. In the context of RDPs, initial work with a Mealy machine representation shows promise [1]. In addition, temporal logic allows for declarative and intuitive models, hence in terms of explainability in RL many possibilities are left, and only some work is just emerging [19].

3 Approach and Software Design

In this paper we develop a new *tool chain* for the recently introduced RDPs and experiment with algorithmic variations, specifically applied to grid worlds (cf. [21]). Figure 2 graphically shows a simplified high-level overview of how decision processes, temporal logic and model checking intertwine. Currently no software tool can conveniently model, visualize and solve all RDPs, which motivates our particular approach. Secondly, RDPs are introduced very recently and not much empirical evidence has been gathered so far [1, 10]. Also the familiar grid worlds in general are underrepresented in the temporal logical RL community despite their abundance in basic RL research, and despite their ability to quickly show insight into models. In general we follow the main paths through Figure 2, where rectangles, diamonds and circles represent formalisms (or models), processes and artifacts, respectively. On the left we see temporal logics such as LTL_f / LDL_f used to define non-Markov decision models, as we have seen in the previous sections, where we also described how these can be compiled into (extended) MDPs, which can then be solved by traditional MDP algorithms. Note that NMDPs are not solely dependent on temporal logic, but require other input such as a state space definition. In the lower-most flow, temporal formulae can be used to define

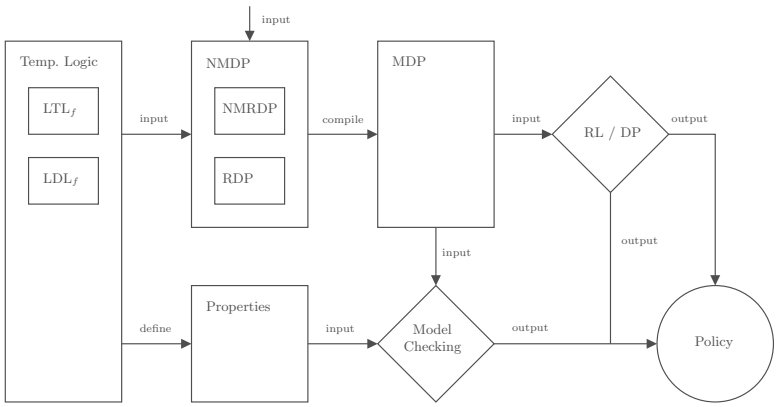


Fig. 2: Conceptual/Tool chains

system properties that can be verified using *model checking* [1]. Finally, experimental learning algorithms can be combined with formal verification methods to produce a policy.

Our prototype integrates existing software tools. First, an integration is made with FLLOAT [6], a tool that allows to construct automata from LTL$_f$ and LDL$_f$ formulae. Because the prototype is a TypeScript (TS) web application, and FLLOAT is built with Python, a small web server is put in place to communicate with FLLOAT. Communication then occurs by making HTTP requests from the prototype through a *Browser HTTP Layer* to the FLLOAT application through a *Server API Layer*. In addition, an integration with a browser-based Graphviz extension called Viz.js[2] was made. It allows for *visualization* of automata within the prototype. Input to the software prototype is defined in terms of TS variables, stored in a single TS file an presented during execution at runtime.

3.1 Compilation: from RDP to MDP

A core component in our approach is the conversion of NMDPs to MDPs for both *off-line*, i.e. before learning, and *on-line*, i.e. during learning, use cases. Intuitively, one can think of the off-line case as a model-based control problem, where the reward function and transition function are fully known to the agent. However, in contrast to other work, we compute the extended MDP *incrementally*. On the other hand, the on-line case can be thought of as a model-free control problem where the agent has to interact with the environment to learn an optimal behavior. Also here the algorithm constructs the extended MDP incrementally, but now only in the areas of the state-action space that are actually

[1] In the current paper there is no room to highlight it, but model-checkers such as Storm (https://www.stormchecker.org/) can be employed for shaping and shielding purposes (and more) in this tool chain, cf. [21]).

[2] https://github.com/mdaines/viz.js

Algorithm 1: NMDP to MDP (off-line)

input : NMDP $M = \langle S, A, T, R \rangle$ with LDL_f reward automata Q_i^R and LDL_f transition automata Q_j^T, with $Q_k \leftarrow Q_i^R \cup Q_j^T$ for convenience
output: Extended MDP $M' = \langle S', A', T', R' \rangle$

1 $t \leftarrow 0$; $s'_t \leftarrow s'_0 \leftarrow (q_{1,0}, q_{2,0}, \ldots, q_{k,0}, s_0)$; $A' \leftarrow A$; $S, T, R \leftarrow \emptyset$
2 **while** $s'_t \notin S'$ **do**
3 $s_t \leftarrow \tau(s'_t)$
4 **for** $a \in A(s_t)$ **do**
5 $s_{t+1} \leftarrow T(\mathcal{L}(s_t), a)$
6 **for** $q_{k,t} \in Q_{k,t}$ **do**
7 $q_{k,t+1} \leftarrow \texttt{transition}(q_{k,t}, \mathcal{L}(s_{t+1}))$
8 $s'_{t+1} \leftarrow (q_{1,t+1}, q_{2,t+1}, \ldots, q_{k,t+1}, s_{t+1})$
9 $S' \leftarrow S' \cup \{s'_{t+1}\}$
10 $T'(s'_t, a, s'_{t+1}) \leftarrow T(\mathcal{L}(s_t), a, \mathcal{L}(s_{t+1}))$
11 $R'(s'_t, a, s'_{t+1}) \leftarrow \texttt{sum\_accept}(Q^R_{i,t+1})$
12 $s'_t \leftarrow s'_{t+1}$
13 **return** M'

experienced by the agent in the interaction with the environment. In addition, in this model-free setting it is assumed that the agent has access to only the states of the automata tracking the formulae, just like in other works (e.g. [18]). Throughout the algorithms, automata for rewards are indicated by Q_i^R and automata for transitions are indicated by Q_j^T and their union is denoted Q_k.

The compilation of an NMDP can exploit knowledge of the known dynamics/reward model. Algorithm 1 outlines our algorithm, generalized to NMDPs. It incrementally builds an extended MDP *off-line* by incorporating all LDL_f automata such that only reachable states are generated. Here, off-line means the compilation is done before solving the final MDP. The transition function in Algorithm 1 on Line 10 abstracts away the different transition dynamics between NMRDPs and RDPs by using labelling function $\mathcal{L}$, making it applicable to both models. Furthermore, the state space generated by Algorithm 1 is *minimal* because it only generates states that are reachable, and thus solution algorithms do not waste time on irrelevant states. The resulting MDP can be solved using e.g. *value iteration*, cf. [21].

Because in the model-free setting the reward function and transition dynamics are not known a priori, compilation cannot occur in a similar fashion as in Algorithm 1. We employ a different, *online, incremental* approach in Algorithm 2. Similar to the off-line algorithm it *incrementally* builds the extended MDP, only here the automata Q_i^R for rewards and automata Q_j^T for transitions are *not* known to the agent. Hence, Q_k is not defined as input like it is for Algorithm 1. Furthermore, the extended MDP is not fully defined in terms of dynamics of transitions and rewards. This, in turn, requires an environment capable of handling LDL_f automata for rewards and transitions. In addition to Algorithm 2, a step function first gets the current state s_t from the environment using

Algorithm 2: NMDP to MDP (on-line)

input : Environment `env` with n-step limit per episode, exploration factor ϵ, discount factor γ and `max_episodes`
output: Policy π

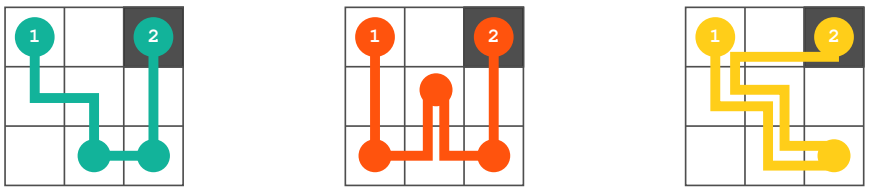
$s_t \leftarrow s_0 \leftarrow$ `env.reset()`; $Q(s,a) \leftarrow$ `arbitrary()` for all $s \in S, a \in A(s)$; $\pi \leftarrow$ `arbitrary()`
repeat
  `ep` $=$ `generate_episode`$(n, A(s_t), \pi, \epsilon)$
  $T \leftarrow |$`ep`$|$
  $G \leftarrow 0$
  foreach *step of* `ep`, $t = T-1, T-2, \ldots, 0$ **do**
    $G \leftarrow \gamma G + r_{t+1}$
    $Q(s_t, a_t) \leftarrow G$
    $\pi(s_t) \leftarrow \arg\max_a(Q(s_t, a))$
until `max_episodes`;
return π

$s_t \leftarrow$ `env.snapshot()`. In addition, all automata are retrieved through $Q_{k,t} \leftarrow$ `env.get_automata_states()`. Then, for each $q_{k,t} \in Q_{k,t}$, both the original state and all automata states transition to their subsequent states through $s_{t+1} \leftarrow$ `env.transition`(s_t, a) and $q_{k,t+1} \leftarrow$ `transition`$(q_{i,t}, \mathcal{L}(s_{t+1}))$ respectively. Automata are then updated through `env.set_automaton_state`$(Q_k, q_{k,t+1})$ the reflect the state transition. Finally, when $Q_{k,t}$ has been iterated over, i.e. all automata have transitioned, a next state is generated by $s'_{t+1} \leftarrow (q_{1,t+1}, q_{2,t+1}, \ldots, q_{k,t+1}, s_{t+1})$, i.e. the MDP state is extended with each *monitor* state. In addition, the rewards for all automata currently in an accepting state are summed by $r \leftarrow$ `sum_accept`$(Q_{k,t+1} \setminus Q^T_{i,t+1})$. Indeed, the better part of Algorithm 2 aligns with first-visit MC [34], except that the underlying problem definition is assumed to be non-Markovian and hence compiled *on-line* from NMDP to MDP.

Similar to Algorithm 1, Algorithm 2 generate only reachable states and is therefore *minimal*. This is due to the transition function of automata being defined as $q_{k,t+1} \leftarrow$ `transition`$(q_{i,t}, \mathcal{L}(s_{t+1}))$, where a transition cannot occur if the target state is unreachable. Due to the nature of RL, the implicitly extended MDP contains only states actually encountered by an agent through interaction with the environment. Note that, as opposed to Algorithm 1, Algorithm 2 does *not* contain all information on the history of states per se. Due to the trial-and-error nature of MC, some states might remain unobserved after Algorithm 2 has completed. Therefore, an optimal policy π^* is only guaranteed in the limit.

4 Experiments

Our experimental evaluation focuses on RDPs for grid worlds, utilizing a model-free online MC algorithm. Our experimental evaluation focuses on RDPs for grid worlds, utilizing a model-free online MC algorithm. Overall, the goal is to



(a) Not so sparse (b) Somewhat sparse (c) Very sparse

Fig. 3: Experiment 1 - Goals for on-line compilation with RL

empirically assess various aspects of RL for RDPs, with a focus on the relation between RDP elements and final learning performance in the resulting extended MDP. For more experiments, (also model-based , value iteration), cf. [21]. In this section we target four different empirical questions: **R1**: *How does learning performance relate to goal sparsity/complexity?*, **R2**: *How can shaping help for complex goals?*, **R3**: *What are the implications of safety properties on learning performance?*, and **R4**: *What is the relation between learning performance and non-Markovian dynamics?*

4.1 Experiment 1: Goal Sparsity

This experiment aims at relating *goal sparsity* to the performance fist-visit MC. Here, goal sparsity describes the accumulated minimum length of traces h_i accepted by LDL_f formulae φ_i. The idea is to increase the grid world size, while keeping reward formulae constant, such that the traces increase in length due to an increase in the size of the state space. To illustrate this, temporal formulae encoding liveness properties are used such that the number of steps to satisfy a formula increases with the grid world size. The minimum length of a trace h_i is measured in terms of the minimum number of states contained in h_i for it to be accepted. The quantitative measurement is defined by the relative frequency of values within 10% of the maximum value. This range is deemed acceptable for a solution as it follows from the value used for ϵ, being $\epsilon = 0.1$, that generates exploration noise. Data is gathered over 50 runs, where each run consists of 1000 episodes with a maximum of 50 steps per episode. Furthermore, $\gamma = 1$. In order to consistently increase the goal sparsity when the grid world size is increased, the agent always starts in state s_{11} and an episode is terminated when the agent reaches s_{13}, after which a new episode is initiated until the maximum number of episodes is reached. Figure 3 outlines three goals, each rewarded +1000, with Figure 3a being the least sparse where goal state are adjacent, Figure 3b being somewhat sparse where goal states require the agent to travel through the center of the grid and Figure 3c being the most sparse where the agent is required to go reach the far-right state and then go back to its initial state again. The goals encoded in LDL_f as $\langle true^*; x_{is2} \wedge y_{is3}; true^*; x_{is3} \wedge y_{is3}; true^* \rangle end$,

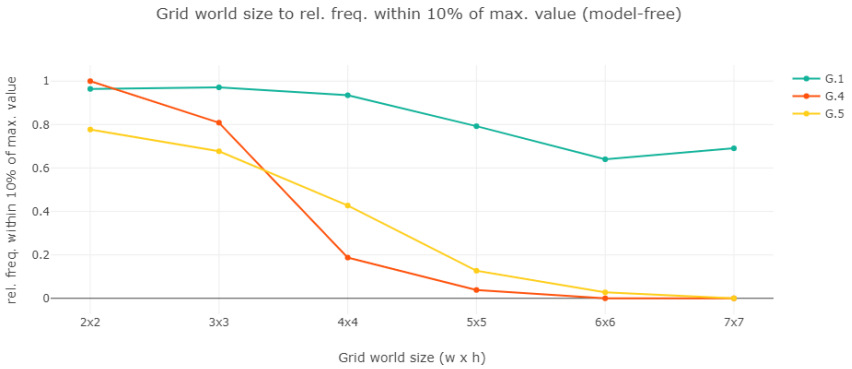


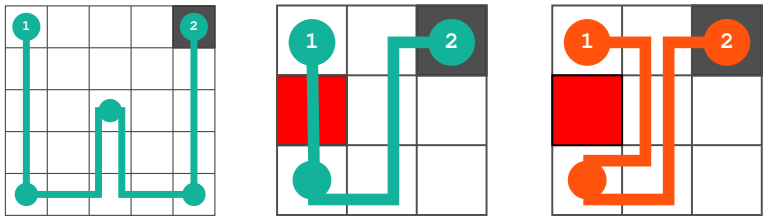
Fig. 4: Rel. freq. within 10% of max. value

$\langle true^*; x_{is1} \wedge y_{is3}; true^*; x_{is2} \wedge y_{is2}; true^*; x_{is3} \wedge y_{is3}; true^* \rangle end$ and $\langle true^*; x_{is3} \wedge y_{is3}; true^*; x_{is1} \wedge y_{is1}; true^* \rangle end$ respectively.

Figure 4 outlines the results for this experiment. It shows that the latter two goals quickly become harder to solve in this setting, as there is a steep decline in optimal behaviour between grid world sizes 3×3 and 4×4. In addition, the graph shows that for the more sparse goals the chance of finding optimal behaviour under the conditions outlined for this setting for a grid world of 7×7 becomes nil. Thus, the observed data indicates there is a relation between goal sparsity and the performance of first-visit MC.

4.2 Experiment 2: Reward Shaping

Recall that for Experiment 1, sparser goals quickly become harder to solve. Therefore, this experiment aims to apply RS to an RDP construction from Section 3 so as to identify whether the performance of MC can be improved when using a potential function from Section 2.4. In addition, Algorithm 2 will be used for on-line compilation in a model-free setting. A preliminary experiment showed that for a 5×5 grid world, in which $\langle true^*; x_{is1} \wedge y_{is5}; true^*; x_{is3} \wedge y_{is3}; true^*; x_{is5} \wedge y_{is5}; true^* \rangle end$ is used, an optimal policy is rarely found [21]. Therefore, this experiment outlines the effect of applying a potential function for RS. For this experiment, a 5×5 grid world is used, transitions are deterministic and MC is applied as the RL learning algorithm. The reward for the goal is set to $+1000$. Figure 5a outlines a possible optimal trace for the given goal. The quantitative measurements are defined by the averaged returns per episode and the size of the extended MDP. A total of 50 runs with each 1000 episodes and a maximum of 50 steps per episode is used. Parameters are defined as $\gamma = 1$ and $\epsilon = 0.1$. The agent always starts an episode in state s_{11} and an episode is terminated when the agent reaches s_{51}, after which a new episode is initiated until the maximum number of episodes is reaches.



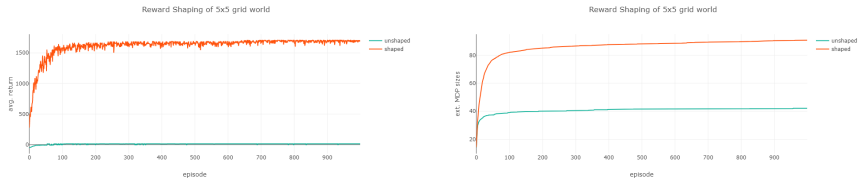
(a) Shaping experiment (b) Unsafe (c) Safe

Fig. 5: (a) Possible optimal trace for reward shaping experiment, (b)-(c) Possible optimal traces for safety experiment

Figure 6 plots the results for this experiment. From Figure 6a it can be observed that a shaped reward takes somewhat longer to learn, but the averaged return is significantly higher for the 5×5 grid world. More specifically, the averaged return for unshaped rewards shows that unshaped rewards are, on average, not received, as the trend remains just below zero. Finally, Figure 6b outlines that shaped rewards reach far more states when compared to unshaped rewards. Given the observation that unshaped rewards are, on average, not received, all states reachable only after a goal is satisfied are very rarely explored for unshaped rewards. Hence, the size of the extended MDP is significantly smaller.

As an intuitive evaluation of the observed results, recall the 5×5 grid world as outlined in Figure 5a. Finding a trace that follows the critical path for the given goal without a potential function is then inherently hard. Consider, for example, the trace outlined by Figure 5a. This trace contains 16 consecutive steps, where the agent may stray from the path at each one of these steps with a probability ϵ. Even if the agent reaches the end of the trace in the case of unshaped rewards, the back-propagation of the reward value may be insignificant when it updates the state-action value of state s_{11}, as the agent only gets rewarded for the trace when it finally reaches terminal state s_{51}. In turn, on average, the agent might only obtain the unshaped reward relatively rarely, leaving most of the states that account for the latter part of the trace uncharted. Note that, as discussed, this result is accounted for in Figure 6b. Conversely, a shaped reward encourages the agent to better follow the critical path, in turn increasing the probability of satisfying the reward formula by its trace and thus increasing the number of explored states in the extended MDP generated by Algorithm 2. In general, it can be observed that shaped rewards make learning perform better, while at the same time increasing the size of the (encountered) state space and the number of steps required in optimal traces.

When reward shaping is applied, a significant increase in MC performance can be observed from Figure 7 for a 5×5 grid world. Where the unshaped reward decreases rapidly between 3×3 and 4×4 grid world, the shaped reward decreases significantly less over the course of the increasing grid world sizes.



(a) avg. return (b) ext. MDP size

Fig. 6: MC performance for unshaped vs. shaped rewards.

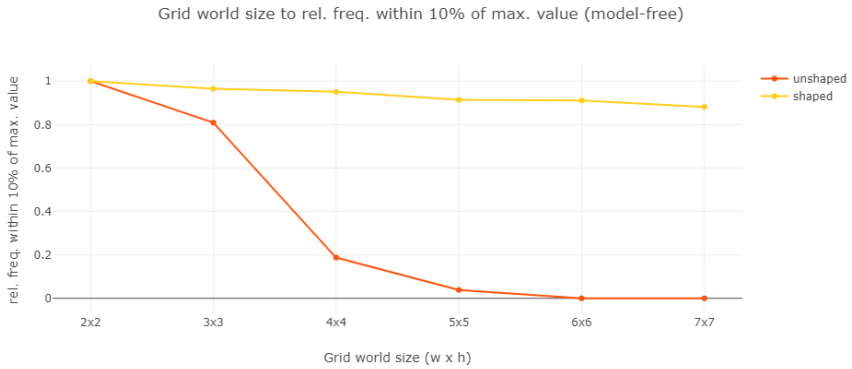
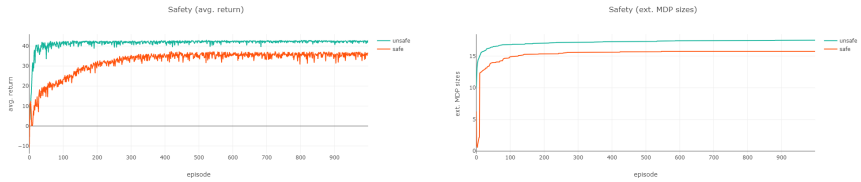


Fig. 7: Rel. freq. within 10% of max. value for unshaped and shaped rewards

4.3 Experiment 3: Safety

The goal of this experiment is to empirically measure the effects of safety properties on the performance of learning. Here, a preemptive shield [2] is applied such that the agent is provided a list of safe actions upon action selection. Moreover, on-line compilation as outlined in Algorithm 2 is applied. Next, a 3×3 grid world is used and modeled as an RDP in which first-visit MC is applied as an RL learning algorithm. Transitions are deterministic to reduce experiment complexity. The quantitative measurement is defined by the performance of the learning algorithm. A total of 50 runs is performed, each of which consists of 1000 episodes and a maximum of 50 steps per episode. Furthermore, $\gamma = 1$ and $\epsilon = 0.1$. The agent always starts in state s_{11} and an episode always terminates when the agent reaches state s_{31}. Goal $\langle true^*; x_{is1} \wedge y_{is3}; true^* \rangle end$ is specified for which the agent is rewarded $+50$ for reaching state s_{13}, i.e. the bottom-left state. A step cost of -1 is rewarded with each step the agent takes in the environment. An additional safety property $[true^*]\langle(\neg x_{is1} \wedge \neg y_{is2})^* \rangle end$ is specified in LDL_f where the agent should never visit unsafe state s_{21}. Figure 5 outlines possible optimal traces for unsafe and safe situations in Figure 5b and Figure 5c



(a) avg. return (b) ext. MDP size

Fig. 8: MC performance for safety

respectively. Note that terminal states, i.e. the top-right states in both Figures 5b and 5c, are marked solid black and that unsafe states, i.e. the center-left states in both Figures 5b and 5c are marked solid red.

First, in order to verify no unsafe condition is met, Figure 9 outlines the extended MDP for this experiment. Because of a technical index mapping, an unsafe state would have a label that starts with $(1, 2, \ldots)$, corresponding to unsafe state s_{12}. As can be observed, there is no state $s' \in S'$ of extended MDP M' for which $\tau(s') \rightarrow s_{21}$. In other words, the unsafe state is never encountered. Therefore, safety property $[true^*]\langle(\neg x_{is1} \wedge \neg y_{is2})^*\rangle end$ is never violated. Furthermore, Figure 8 plots the results for this experiment. Let us reconsider the results from Figure 8a and Figure 8b. It can be observed that, when learning performance is decreased, the size of the state space has become smaller. However, the intuition is that, when less states are to be explored, performance is generally increased. It appears, then, that RL performance is not necessarily dictated by the size of the state space. To account for what does impact the decreased learning performance, let us reconsider the experiment setup. Where state s_{13} in Figure 5b can be reached from two states, i.e. s_{12} and s_{23}, the same state can only be reached from a single state s_{23} in Figure 5c. This observation leads to the conjecture that RL performance is related to the *accessibility* of states required to satisfy goal formulae. That is, when states have less paths by which they can be reached, RL performance decreases.

4.4 Experiment 4: Non-Markovian Transitions

This experiment focuses on non-Markovian transition models. Here, on-line compilation using Algorithm 2 will be used in a model-free setting, with first-visit MC. The quantitative measurements are defined by the averaged returns per episode, the averaged number of steps per episode and the size of the extended MDP. The experiment consists of 50 runs, each 1000 episodes with a maximum of 50 steps per episode. Furthermore, $\gamma = 1$ and $\epsilon = 0.1$. The grid world is defined by a 5×2 rectangle. The agent always starts an episode in state s_{11} and state s_{51} is defined as a terminal state. There is a single goal $\langle true^*; x_{is3} \wedge y_{is1}; true^*\rangle end$ rewarded when reaching s_{41} of $+10$, with the addition of a step cost encoded in LDL_f valued as -1.

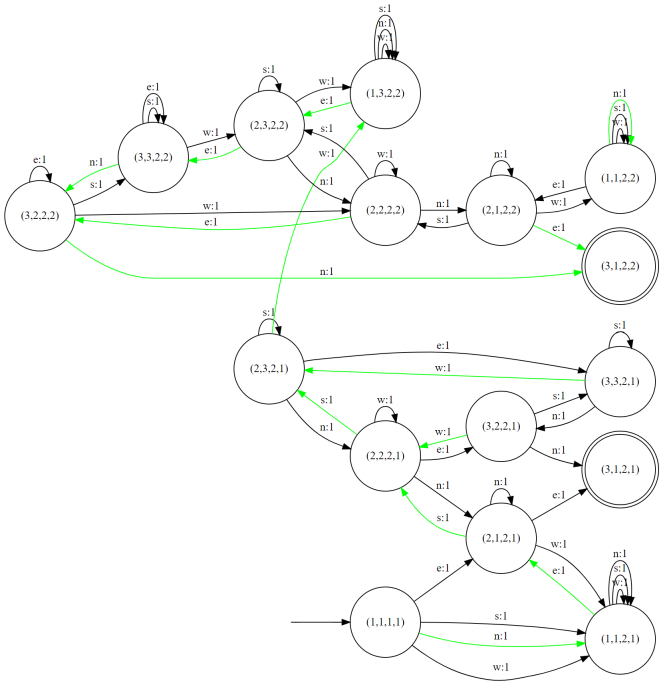
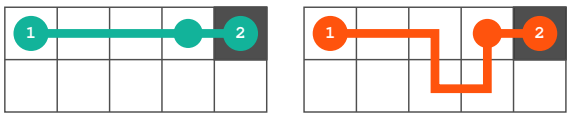


Fig. 9: Product MDP for safety experiment

The first set of 50 runs uses non-deterministic transitions. That is, in every state of the grid world the agent has a 0.8 probability of ending up in the next state and a 0.2 probability of remaining in its current state. For example, when in s_{21} and taking action s, there is a 0.8 probability that we end up in s_{22} and a 0.2 probability to remain in s_{21}. The transition from s_{31} for action e is defined by $\langle true^*; x_{is3} \wedge y_{is1} \rangle end$ and will be different for the regular transition defined below. A possible optimal trace is visually displayed in Figure 10a.

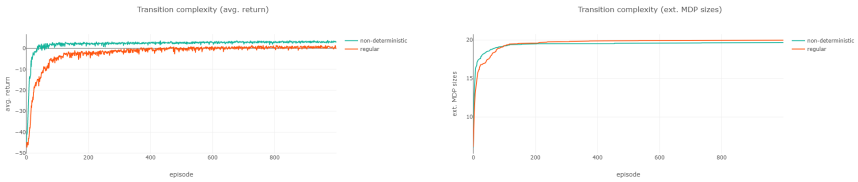
The second set of 50 runs uses regular transitions. Now, when the agent reaches s_{31}, a transition $\langle true^*; x_{is2} \wedge y_{is1}; x_{is3} \wedge y_{is1} \rangle end$ is defined that depends on whether the agent came from state $_{21}$ or not. In case the previous state was s_{21}, the transition of action e in s_{31} is defined as a 0.1 probability of ending up in s_{41} and a 0.9 probability of remaining in s_{31}. Otherwise, the same transition probabilities used for the non-deterministic transitions apply, i.e. a 0.8 probability of ending up in s_{41} and a 0.2 probability of remaining in s_{31}. Here, the transition in s_{31} is regularly defined and depends on a history of states. A possible optimal trace for regular transitions is given in Figure 10b.

From Figure 11, we observe that regular non-deterministic transitions, when compared to non-regular ones, induce a harder problem for a model-free setting, while the size of the state space is only increased by one additional state that



(a) Non-deterministic (b) Regular

Fig. 10: Possible optimal traces for transition complexity variations



(a) avg. return (b) ext. MDP size

Fig. 11: MC performance for transition complexity variations

keeps track of whether or not s_{21} has been visited. In other words, a small increase in size can evidently generate a significantly harder problem.

5 Conclusions and Future Work

We have introduced a new tool chain to compute with regular decision processes, and experimented with novel algorithmic variations with the aim to gain insight in how complexity of temporal logic formulas relates to the complexity of learning algorithms such as MC RL for the resulting extended MDPs. We have shown that by increasing the world size for similar built formulas problems get harder (**R1**)), but also that reward shaping on the automata representing those formulas can really help learning, *and* exploration (**R2**). The safety experiments (**R3**) have shown less states do not necessarily result in easier learning tasks, and the non-Markov transitions experiments (**R4**) showed that these only caused a small increase in state space size, but did complicate learning a lot more.

Our overall conclusions of the experiments point to our main future work direction. It seems that there are complex relations between i) the complexity and properties of the temporal formulae defining the non-Markovian aspects, ii) the resulting size and connection structure of the extended MDP, and iii) the learning performance of online RL algorithms for the extended MDP. Much more work is needed to evaluate a temporal specification for a particular problem, and assess its influence on the complexity of learning the original task in the presence of the new rule. For MDPs there is much work on measures relating to e.g. homomorphism and abstraction [36, 37] and work is starting to emerge to gain more insight in the logical side [31] but their interaction needs study.

Other future work should focus on *representations* and *applications*. For the first, there is much to be gained by utilizing existing formal methods, for example the use of transducers and Mealy machines [9] trading off the size of the state space with compositional modeling. Equally important is to focus more on utilizing model checking tools [2, 16]. Application-wise, there are plenty of opportunities to utilize the methods in this paper, for example to constrain RL dialogue agents, in medical domains with logically represented medical guidance and regulations, or to implement coaching strategies in RL coaching agents [17].

References

1. Abadi, E., Brafman, R.I.: Learning and solving regular decision processes. In: IJCAI (2020)
2. Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., Topcu, U.: Safe reinforcement learning via shielding. In: AAAI (2018)
3. Bacchus, F., Boutilier, C., Grove, A.: Rewarding behaviors. In: AAAI (1996)
4. Brafman, R., Giacomo, G.D., Patrizi, F.: LTLf/LDLf non-markovian rewards (2018)
5. Brafman, R.I., De Giacomo, G.: Planning for LTLf/LDLf goals in non-markovian fully observable nondeterministic domains. In: IJCAI (2019)
6. Brafman, R.I., De Giacomo, G.: Regular decision processes: A model for non-markovian domains. In: IJCAI (2019)
7. Camacho, A., Icarte, R.T., Klassen, T.Q., Valenzano, R.A., McIlraith, S.A.: LTL and beyond: Formal languages for reward function specification in reinforcement learning. In: IJCAI (2019)
8. Camacho, A., McIlraith, S.A.: Learning interpretable models expressed in linear temporal logic. In: ICAPS (2019)
9. De Giacomo, G., Favorito, M., Iocchi, L., Patrizi, F., Ronca, A.: Temporal Logic Monitoring Rewards via Transducers. In: KR (2020)
10. De Giacomo, G., Iocchi, L., Favorito, M., Patrizi, F.: Reinforcement learning for ltlf/ldlf goals. arXiv preprint arXiv:1807.06333 (2018)
11. De Giacomo, G., Iocchi, L., Favorito, M., Patrizi, F.: Foundations for restraining bolts: Reinforcement learning with LTLf/LDLf restraining specifications. In: ICAPS (2019)
12. De Giacomo, G., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: AAAI (2013)
13. Fulton, N., Platzer, A.: Safe reinforcement learning via formal methods: Toward safe control through proof and learning. In: AAAI (2018)
14. Furelos-Blanco, D., Law, M., Jonsson, A., Broda, K., Russo, A.: Induction and exploitation of subgoal automata for reinforcement learning. Journal of Artificial Intelligence Research **70**, 1031–1116 (2021)
15. Garcıa, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research **16**(1), 1437–1480 (2015)
16. Giaquinta, R., Hoffmann, R., Ireland, M., Miller, A., Norman, G.: Strategy synthesis for autonomous agents using PRISM. In: NASA Formal Methods Symposium. pp. 220–236. Springer (2018)
17. el Hassouni, A., Hoogendoorn, M., van Otterlo, M., Barbaro, E.: Personalization of health interventions using cluster-based reinforcement learning. In: International Conference on Principles and Practice of Multi-Agent Systems (2018)

18. Jothimurugan, K., Alur, R., Bastani, O.: A composable specification language for reinforcement learning tasks. In: NeurIPS (2019)
19. Kasenberg, D., Thielstrom, R., Scheutz, M.: Generating explanations for temporal logic planner decisions. In: ICAPS (2020)
20. Kim, J., Muise, C., Shah, A., Agarwal, S., Shah, J.: Bayesian inference of linear temporal logic specifications for contrastive explanations. In: IJCAI (2019)
21. Lenaers, N.: An Empirical Study on Regular Decision Processes for Grid Worlds. Master's thesis, Department of Computer Science, Faculty of Science, Open University (2021)
22. Li, X., Vasile, C.I., Belta, C.: Reinforcement learning with temporal logic rewards. In: IROS (2017)
23. Liao, H.C.: A survey of reinforcement learning with temporal logic rewards (2020)
24. Liao, S.M.: Ethics of artificial intelligence. Oxford University Press (2020)
25. Littman, M.L., Topcu, U., Fu, J., Isbell, C., Wen, M., MacGlashan, J.: Environment-independent task specifications via GLTL. arXiv preprint arXiv:1704.04341 (2017)
26. Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J.W., Songhori, E., Wang, S., Lee, Y.J., Johnson, E., Pathak, O., Nazi, A., et al.: A graph placement methodology for fast chip design. Nature **594**(7862) (2021)
27. Ng, A.Y., Harada, D., Russell, S.J.: Policy invariance under reward transformations: Theory and application to reward shaping. In: ICML (1999)
28. van Otterlo, M.: Ethics and the value (s) of artificial intelligence. Nieuw Archief voor Wiskunde **5**(19), 3 (2018)
29. Pnueli, A.: The temporal logic of programs. In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science (1977)
30. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley (1994)
31. Romeo, Í.Í., Lohstroh, M., Iannopollo, A., Lee, E.A., Sangiovanni-Vincentelli, A.: A metric for linear temporal logic. arXiv preprint arXiv:1812.03923 (2018)
32. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature **529**(7587) (2016)
33. Spaan, M.T.: Partially observable markov decision processes. In: Wiering, M.A., van Otterlo, M. (eds.) Reinforcement Learning, pp. 387–414. Springer (2012)
34. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press (2018)
35. Thiébaux, S., Gretton, C., Slaney, J., Price, D., Kabanza, F.: Decision-theoretic planning with non-markovian rewards. JAIR **25** (2006)
36. Van Otterlo, M.: The Logic of Adaptive Behavior, Frontiers in Artificial Intelligence and Applications, vol. 192. IOS Press, Amsterdam (2009)
37. Wang, H., Dong, S., Shao, L.: Measuring structural similarities in finite mdps. In: IJCAI (2019)
38. Wang, H.n., Liu, N., Zhang, Y.y., Feng, D.w., Huang, F., Li, D.s., Zhang, Y.m.: Deep reinforcement learning: a survey. Frontiers of Information Technology & Electronic Engineering (2020)
39. Wiering, M.A., Van Otterlo, M.: Reinforcement learning, Adaptation, learning, and optimization, vol. 12. Springer (2012)

Implementation of a Distributed Minimum Dominating Set Approximation Algorithm in a Spiking Neural Network

Victoria Bosch[⋆1][0000−0001−7454−8325], Arne Diehl[⋆1][0000−0001−9702−1083], Daphne Smits[⋆1][0000−0002−3737−6672], Akke Toeter[⋆1][0000−0002−9577−920X], and Johan Kwisthout[2][0000−0003−4383−7786]

[1] School for Artificial Intelligence, Radboud University, Montessorilaan 3 6525 HR Nijmegen, the Netherlands
[2] Donders Center for Cognition, Radboud University, Montessorilaan 3 6525 HR Nijmegen, the Netherlands
j.kwisthout@donders.ru.nl

Abstract. Neuromorphic computing is a promising new computational paradigm that may provide energy-lean solutions to algorithmic challenges such as graph problems. In particular, the class of distributed algorithms may benefit from translation to spiking neural networks. This work presents such a translation of a distributed approximation algorithm for the minimum dominating set problem, as described by Kuhn and Wattenhofer (2005), to a spiking neural network. This translation shows that neuromorphic architectures can be used to implement distributed algorithms. Subcomponents of this implementation, such as the calculation of the minimum or maximum of two numbers and degree of a node, can be reused as foundational building blocks for other (graph) algorithms. This work illustrates how leveraging neural properties for the translation of traditional algorithms relies on novel insights, thereby contributing to a growing body of knowledge on neuromorphic applications for scientific computing.

Keywords: Neuromorphic Computing, Spiking Neural Network, Distributed Computing, Minimum Dominating Set, Graph Algorithms

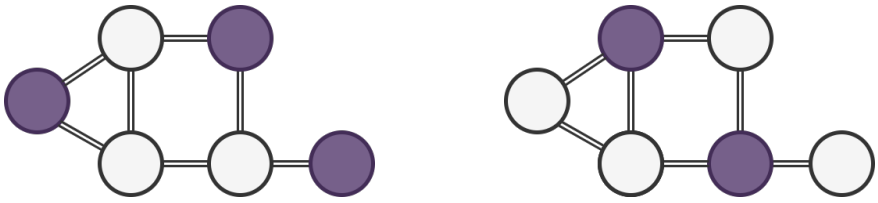
1 Introduction

Neuromorphic Computing is a relatively young field that concerns itself with emulating the brain and bringing advantages that are inherent to its structure into computational devices and programs. These advantages include parallel architecture, co-location of memory and computation, and high energy efficiency [14]. Moving away from the traditional von Neumann-architecture is an avenue from which various areas of research and development could benefit. In particular, the use of neuromorphic architectures has prompted the development of new methods and algorithms for scientific computing [16].

⋆ equal contribution

The basic architecture and computational model underlying neuromorphic hardware [4, 2] is the spiking neural network (SNN). SNNs offer a great potential in finding solutions to graph problems. The translation process of graph algorithms to SNNs is relatively new, and there is still much knowledge to be gained about the optimal conversion and optimisation. The structure of SNNs (neurons and synapses) is similar to that of graphs (vertices and edges), allowing any graph to be represented by a spiking neural network. Thus, one possible approach is to use the inherent structure of the graph to solve various problems by letting the vertices communicate with spikes and spike timing (message-passing). Recent work that takes this approach is the partial translation of an algorithm for the max network flow problem to neuromorphic hardware [2], implementation of a SAT solver [21], and an exploration of neuromorphic algorithms for the longest shortest path and minimum spanning tree [9]. Another approach for the usage of SNNs for graph problems is to view every neuron as a computational unit. Manually programming and designing the network may be unconventional, yet this approach enables increased control in tailoring of SNNs for various (graph) problems. For example, Aimone et al. present a conversion method for the class of dynamic programs [1].

However, there are more classes of algorithms that may benefit from neuromorphic architectures, especially the class of distributed algorithms. This is because distributed computing traditionally requires multiple CPUs, whereas the neurons in an SNN can function as a population of computational units within one device.



(a) Dominating set (b) Minimum Dominating Set

Fig. 1: *Examples of a dominating set and a minimum dominating set of a graph (shown in purple).*

To demonstrate the potential of programming SNNs for distributed graph algorithms, we show a conversion of a distributed approximation algorithm for the *minimum dominating set* (MDS) graph problem. Kuhn and Wattenhofer [11] have constructed a distributed and constant-time approximation algorithm for the MDS problem. The MDS of a graph is a smallest subset of the vertices in G, such that for every vertex it either is in the dominating set, or one of its direct neighbours is (see fig. 1). The Kuhn-Wattenhofer algorithm is a parallelised greedy algorithm and achieves an expected approximation ratio of $\mathcal{O}(k\Delta^{(2/k)}log\Delta)$ in $\mathcal{O}(k^2)$ rounds where k is an arbitrary parameter that denotes the number of iterations of the approximation algorithm and Δ is the maximum

degree in graph G. Because of its distributive nature and constant-time complexity, we have found the algorithm fit to be effectively implemented in a spiking neural network. Thus, we present a spiking neural network implementation of the Kuhn-Wattenhofer approximation algorithm for the minimum dominating set problem, in order to show the potential of programming SNNs.

2 Preliminaries

2.1 Graph Definitions

A graph $G = (V, E)$ with edges $(u, v) \in E$, consists of vertices $V = (v_1, ..., v_n)$ and edges $E = (e_1, ..., e_m)$, where n and m represent the number of vertices and edges in graph G respectively. The degree δ_i represents the number of connected vertices for an arbitrary vertex v_i with $i \in [1, n]$. Alternatively, $\delta_i^{(1)}$ and $\delta_i^{(2)}$ denote the maximum degree in a one- and two-step neighbourhood of the vertex v_i respectively. Δ denotes the maximum degree in the graph G.

2.2 Minimum Dominating Set

The functional *minimum dominating set problem* is defined as follows:

> **Minimum Dominating Set**
> **Input:** Undirected graph $G = (V, E)$.
> **Output:** Subset D in which $D \subseteq V$ if $v \in V$ is in D or adjacent to D, and no subset of D is a dominating set of G.

The minimum dominating set problem has historical roots in the k-queens problem and is related to the set cover problem.The set cover problem can be considered equivalent to MDS under L-reduction [7]. The MDS problem is one of the first graph problems shown to be NP-hard [5]. The best logarithmic approximations of the MDS are achieved by hybrid algorithms that make use of greedy algorithms and LP-rounding [13]. For a recent overview of the performance of various approximation algorithms for the MDS problem, we refer to [13]. Potential applications for algorithms that solve the minimum dominating set problem include the clustering of wireless devices in a network [8] and automatic text summarisation [20].

2.3 Neural Model

Here we define a spiking neural network as a finite directed graph with vertices and edges, where the vertices are neurons and the edges function as synapses. We make use of the leaky-integrate-and-fire (LIF) neural model, which is commonly used in neuromorphic hardware. A LIF-neuron is defined by its initial voltage (V_{init}), the activation threshold (thr), the amplitude of the spike that occurs when threshold is met (amp), the leakage constant (m) which decreases the voltage over time, and the reset voltage to which the neuron returns after

spiking (V_{reset}). Neurons can either be deterministic or stochastic, which determines their spiking behaviour. A deterministic neuron spikes when its voltage has reached its threshold. A stochastic neuron however, will spike according to some probability distribution p. Neurons are connected by synapses, which are defined by the pre- and post-synaptic neuron that they connect, the weight (w) of their connection, and the time delay (d) of the signal. Spiking neural networks can take input from various sources, for example, the programmed V_{init} of a neuron, or from neurons that are programmed to spike at a certain time. The graphical notation for spiking neural networks in this paper is defined in fig. 2 and is based on the notation presented in [2].

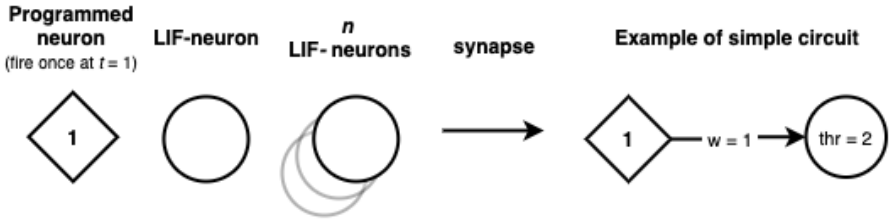


Fig. 2: *Notation for the graphical representation of spiking neural networks. If the values are set to default values, they are omitted from the shown graph.*

2.4 Kuhn-Wattenhofer algorithm

The Kuhn-Wattenhofer algorithm consists of two parts. It commences by solving the LP-relaxation of the problem (in alg. 2), and then uses the solution, the α-approximation ($\underline{x}^{(\alpha)}$) of LP_{MDS}, to approximate the integer program (in alg. 1). Using distributed randomised selection, a solution $\underline{x}_{DS}$ for the integer program (IP_{MDS}) is found, where $\underline{x}_{DS}$ consists of a binary list indicating which vertices are in the dominating set that approximates the minimum dominating set.

The approximation of LP_{MDS} (alg. 2) contains the main functionality of the Kuhn-Wattenhofer algorithm, as it returns the approximation of the related linear programming solution LP_{MDS}. It is a distributed greedy algorithm, where each vertex v_i dynamically calculates the α-approximation x_i for the solution to LP_{MDS}. To that end, each vertex also has a variable *colour*, which is initially white and turned grey if the vertex is considered covered. Each vertex also has a variable dynamic degree $\tilde{\delta}(v_i)$, which is equal to the number of vertices in the closed neighbourhood (that includes the vertex itself) that are white. As initially all vertices are white, the dynamic degree is initialised to the number of vertices in its closed neighbourhood ($\delta_i + 1$). The algorithm contains two nested loops. For every iteration, the vertices with a dynamic degree $\tilde{\delta}(v_i)$ above the threshold, raise x_i. Next, the dynamic degree $\tilde{\delta}(v_i)$ is updated according to the

Algorithm 1: Kuhn-Wattenhofer - $LP_{MDS} \rightarrow IP_{MDS}$

input : feasible solution $\underline{x}^{\alpha}$ for LP_{MDS}
output: IP_{MDS}-solution $\underline{x}_{DS}$ (dom. set)

calculate $\delta_i^{(2)}$ // Each step is computed for all vertices $v_i \in V$ simultaneously.
$p_i := min\{1, x_i^{\alpha} \cdot ln(\delta_i^{(2)} + 1)\}$
$x_{DS,i} := \begin{cases} 1 & \text{with probability } p_i \\ 0 & \text{otherwise} \end{cases}$
send $x_{DS,i}$ to all neighbours
if $x_{DS,j} = 0$ *for all* $j \in N_i$ **then**
  $x_{DS,i} := 1$
end

neighbouring colour values, which are then updated according to the x values of neighbouring vertices v_i.

These two algorithms work under the assumption that all vertices have knowledge of the maximum degree Δ. There is a third algorithm available in [11], which describes an adaptation of the LP_{MDS} approximation in which this knowledge is not assumed. However, for the scope of this research, only the first two algorithms are implemented. For a more detailed explanation of the workings of the Kuhn-Wattenhofer algorithm, proofs, and the third algorithm, we refer to the original paper [11].

Algorithm 2: Kuhn-Wattenhofer - LP_{MDS} approximation

$x_i := 0$;
$\tilde{\delta}(v_i) := \delta_i + 1$
for $l :=$ *k-1 to 0 by -1* **do**
  for $m :=$ *k-1 to 0 by -1* **do**
    if $\tilde{\delta}(v_i) \geq (\Delta + 1)^{l/k}$ **then**
      $x_i := max\{x_i, \frac{1}{(\Delta+1)^{m/k}}\}$
    end
    Send $colour_i$ to all neighbours
    $\tilde{\delta}(v_i) :=\mid \{j \in N_i \mid colour_j = \text{'white'}\} \mid$
    Send x_i to all neighbours
    if $\Sigma_{j \in N_i} x_j \geq 1$ **then**
      $colour_i :=$ 'gray'
    end
  end
end

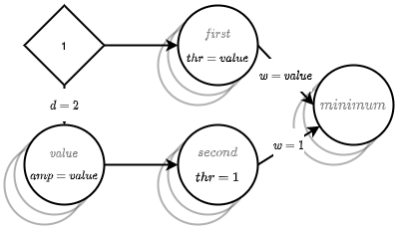
3 Spiking Implementation

In this section, we present the spiking implementation of the Kuhn-Wattenhofer algorithm and the details of the various functions that enable the calculation. The original algorithm can be viewed in section 2.4. The spiking implementation consists of multiple spiking neural networks, that each handle specific calculations and implement different functionalities. Some of these functions are called multiple times, such as the calculation of the degree of the neurons. Programming the SNN is achieved by setting and defining the variables of the neurons, such as their thresholds, delays and spiking amplitude, and the weights of the synapses. The values are defined by e.g. information acquired from the input graph structure, and several networks take the measured resulting voltage of another network as input.

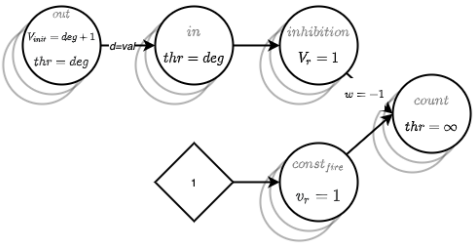
3.1 LP-relaxation

The spiking implementation of the LP-relaxation (alg. 1) consists of six spiking networks. The first function, *spiking_degree*, calculates the degree of each neuron. This is done by creating neurons for all vertices and bidirectional synapses for all edges. All neurons spike once, and the resulting voltage of each neuron then represents the degree of that neuron. Then, *spiking_max_degree*, calculates the maximum value of each neurons neighbourhood. The constructed network of this function can be seen in fig. 3b. It is implemented by creating an *out* and *in*-neuron for each neuron, where for each edge, a synapse is created between the *out*-neuron and the *in*-neuron. The *out*-neurons spike once, with a delay equal to their value. The *in*-neurons have a threshold equal to their degree and no leakage, which ensures that they spike when the last spike has arrived. The spike-timing of the *out*-neuron therefore represents the maximum value of the neighbourhood. This is then converted to a voltage-representation using a separate *count* neuron, which adds one to its voltage until the *in*-neuron fires.

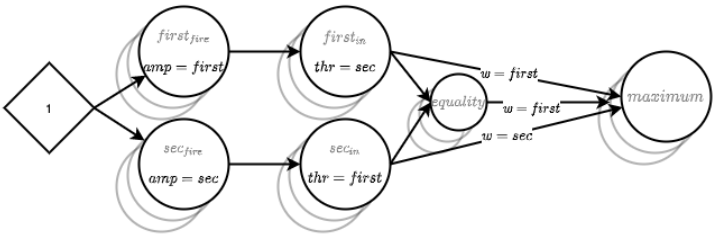
Afterwards, *spiking_multiplication*, calculates an element-wise multiplication of two arrays. This is implemented using one synapse per element, where the initial spike amplitude represents the first value and the synaptic weight represents the second value. The voltage of the post-synaptic neurons then represents the result of the multiplication. Then, the *spiking_minimum* network calculates the element-wise minimum of an element in an array and 1. The constructed network of this function can be seen in fig. 3a. The minimum is calculated using the following function: $((1 > value) \cdot value) + ((value > 1) \cdot 1)$. The neuron *first* handles the first condition $(1 > value)$ by receiving an input spike of amplitude 1 and having a threshold equal to the provided value. The neuron *second* handles the second condition $(value > 1)$ by receiving an input spike of amplitude equal to the value and having a threshold of 1. Both neurons are connected to a final neuron *minimum*, with a synaptic weight equal to *value* for the first neuron and 1 for the second. The voltage of this last neuron then represents the minimum value.



(a) *Spiking_min: Network for calculating the minimum of 1 and the incoming value.*



(b) *Spiking_max_degree: Network for calculating the maximum degree of a neurons neighbourhood.*



(c) *Spiking_max: Network for calculating the maximum of two values.*

Fig. 3: *Various spiking neural networks used as modules in the spiking implementation of the Kuhn-Wattenhofer algorithm.*

The fifth function, *spiking_sampling*, samples according to the given probabilities. This is implemented by creating a stochastic neuron for every vertex, which spikes with the given probabilities. The spike represents whether a neuron is considered in the dominating set or not.

And lastly, the *spiking_summation* network checks for all neurons whether one of their neighbours is in DS, and adds the neuron v_i to DS if this is not the case. This is accomplished by creating neurons for all vertices, and bidirectional synapses for all edges. Each dominating set vertex is represented as a programmed neuron that is constantly spiking. All other vertices are represented as LIF-neurons with a threshold of 1 and a constant input voltage of 1, which initiates them to spike constantly. The weight of the synapse is negative (-1) if the presynaptic neuron is in the dominating set, and 0 otherwise. This way, the LIF-neurons that have a neighbour in the dominating set will be inhibited, while the programmed neurons always keep spiking. The spikes thus represent whether a vertex is considered to be in the dominating set or not.

3.2 Approximating LP_{MDS}

The spiking implementation of the LP_{MDS} approximation (alg. 2) consists of three different spiking networks, of which one is the network utilised in the LP-relaxation to calculate the degree. The second is the *spiking_update* function,

which is depicted in fig. 4 and consists of three main steps. The first step is the updating of the x-values. The *check_dd* neuron checks if the first if-statement is met, by setting its threshold to $(\Delta+1)^{(l/k)}$ and setting the initial voltage to the old dynamic degree ($\tilde{\delta}(v_i)$) values. If this neuron spikes, the x-value is updated. The calculation of the new x-value is done using the *spiking_max* function, which will be described below. The computed x-values are contained in the x neurons and sent to the *check color* neuron. This neuron checks if the second if-statement is met. If this neuron fires, a silencer neuron is activated, which turns the *color* neuron grey by inhibiting it. The dynamic degree is updated by adding the outputs of these *color* neurons. Because the dynamic degree needs to be updated before the *color* neurons are updated, we read out the dynamic degree after step 2 of the simulation. The x and *color* values are saved after three more simulation steps. As indicated in the figure, all of these neurons are created once for every vertex in the input graph.

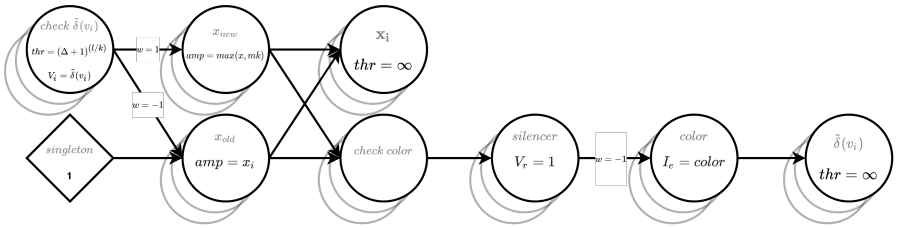


Fig. 4: *Graphical representation of the update function in the approximation of LP_{MDS}. The input values to this function are the previous values of x, color and $\tilde{\delta}(v_i)$ (dynamic degree). These are used in the model as initial voltage V_i, amplitude and input current. The output consists of the voltage levels of the x, color and $\tilde{\delta}(v_i)$ neurons, which are used in the LP-relaxation.*

The *spiking_max* network used in the update function calculates the maximum of two values. The network is depicted in fig. 3c. The maximum is calculated using the following formula: $((a > b)\cdot a)+((b > a)\cdot b)$, and is thus implemented in a similar manner as the *spiking_minimum* network. One addition is made to ensure a correct computation if both values are equal. In that case, the *maximum* neuron spikes and is reset to 0. The *equality* neuron checks if the inputs are equal and sets the *maximum* neuron to the first value.

These functions contain some commonly used principles, such as the implementation of if-else statements using the threshold value of the neurons. For an example, see the descriptions of the *spiking_minimum* and *spiking_maximum* functions.

However, there are a few functions that have not been converted to the SNN, such as the calculation of the natural logarithm. An attempt was made to perform this calculation with a Taylor approximation in an SNN, but this lead

to the need of more standard Python or package functions. Future work may allow for a complete conversion.

3.3 Neuromorphic Implementation

The translation of the Kuhn-Wattenhofer algorithm has been implemented in the SNN Simulator [19], which simulates a spiking neural network. Additionally, the functions for the calculation of the degree of a neuron and the maximum of two numbers have been implemented in the LAVA framework by Intel [6]. In future work, these functions may be ran on the Intel Loihi neuromorphic chip. Code and documentation can be found at [3].

4 Complexity Analysis

In this section we compare the complexity of our implementation with that of the Kuhn-Wattenhofer algorithm.

Traditional computational complexity analysis observes time complexity in terms of the number of operations that are performed and space complexity as amount of utilised memory. As the Kuhn-Wattenhofer algorithm is a distributed algorithm, our complexity analysis of their algorithm follows the measures as defined by Kshemkalyani and Singhal [10]. According to this metric, time and space complexity are computed both per vertex and system-wide. Additionally, the message complexity is computed in terms of message size, number of messages and number of communication rounds.

Because of their novel structure, spiking neural networks require new measures of complexity for their neuromorphic computation [12]. For SNNs, the time complexity can be measured as time to convergence, the space complexity as network size, and energy complexity as the total number of spikes, according to Kwisthout and Donselaar [12]. Because our unconventional implementation makes use of voltage-based computation, a hybrid complexity analysis is performed. The creation of the network is not performed by the network itself, therefore we also separate the network generation from the simulation of the network in this analysis.

4.1 Space Complexity

Space complexity is defined as the amount of memory that is needed for a computation, apart from the input [17]. However, for spiking neural networks it is defined as network size [12]. For this project, we report on both. As we also make use of non-spiking functions, providing the standard space complexity may provide a more accurate picture. If implemented on neuromorphic hardware, the separation between these two measurements may be more relevant, as a choice can be made to make use of an oracle to compute certain functions at times.

Space Complexity of the Spiking Neural Network The space complexity analysis has indicated that $\mathcal{O}(n^2)$ space is needed for the generation of the spiking networks of both the LP-relaxation (alg. 1) and the LP_{MDS}-approximation (alg. 2). The complexity of the SNN implementation depends on the creation of neurons (n) and synapses (n^2). The execution of the SNNs does not take up any space in addition to their creation, as the execution only changes voltage values within the network. Thus, execution has a space complexity of $\mathcal{O}(1)$. The complete program, including SNN generation, execution, and any necessary non-spiking function has a space complexity of $\mathcal{O}(n^2)$ as well.

Space Complexity of the Kuhn-Wattenhofer Algorithm Per vertex, the space complexity of both Kuhn-Wattenhofer algorithms is $\mathcal{O}(1)$, because each vertex stores the dynamic degree ($\tilde{\delta}(v_i)$), *color* and α-approximation (x_i) which requires 3 memory slots per vertex. resulting in a system-wide space complexity of $\mathcal{O}(n)$. The space complexity of the messages is defined by the message size of $\mathcal{O}(log(n))$ and the amount of messages of $\mathcal{O}(n^2)$.

4.2 Time Complexity

The time complexity of an algorithm is traditionally computed as the amount of atomic computational steps needed in relation to the size of the input. In our case, the input consists of a graph and the maximum degree of the graph. This means that we will express the time complexity as a function of vertices in the input graph.

Since we are building and running a discrete time spiking neural network, we have only calculated the time complexity for the building of the network in the way described above. For the execution of the discrete SNN, we assumed that every time step has a constant time complexity, which is reasonable, if the network is run on neuromorphic hardware. Therefore, the amount of steps the networks have to run determine their time complexity. The time to convergence, as suggested by Kwisthout and Donselaar [12], may be more appropriate for decision problems than for the goals of this research.

Time Complexity of the Spiking Neural Network The time complexity of generating the SNNs used in the LP-relaxation is $\mathcal{O}(n^2)$. The main contributor in this time complexity is the calculation of the δ_i and Δ, which both require synapses between neurons (in both directions) for every edge in graph G. The generation of the SNNs used in the LP_{MDS}-approximation has a time complexity of $\mathcal{O}(k^2 \cdot n^2)$. The bottleneck in this generation is formed by the two for-loops and the update function that they contain, in which the dynamic degree $\tilde{\delta}(v_i)$ is calculated, which requires bidirectional edges.

The execution of LP-relaxation has a time complexity of $\mathcal{O}(n)$, which is due to the calculation of $\delta^{(1)}$ and $\delta^{(2)}$, which require Δ time steps. The upper bound and worst case scenario of Δ, the maximum degree of all neurons, here is $\mathcal{O}(n)$.

Execution of the LP_{MDS} approximation has a time complexity of $\mathcal{O}(k^2)$, due to the two for-loops. The final time complexity of the execution is thus $\mathcal{O}(n + k^2)$.

The complete program, including SNN generation, execution, and any necessary non-spiking function, has a time complexity of $\mathcal{O}(k^2 \cdot n^2)$ due to the construction of the SNNs.

Time complexity of the Kuhn-Wattenhofer algorithm Per vertex, the time complexity of the LP-relaxation is $\mathcal{O}(n)$. Computation of $\delta^{(2)}$ and x_{DS} are the main contributors to this complexity. Per vertex, the time complexity of LP_{MDS} approximation is $\mathcal{O}(k^2 \cdot n)$, due to the two for-loops and the computation of the dynamic degree within them. The time complexity of the messages is defined by the number of communication rounds of $\mathcal{O}(k^2)$. Note that Kuhn and Wattenhofer only mention the message time complexity of $\mathcal{O}(k^2)$ in their paper, constituting to their claim of a constant-time algorithm. However, we argue that because the system-wide time complexity is not constant in time, this claim is invalid.

4.3 Energy Complexity

Energy complexity is an uncommon, widely debated and yet undefined complexity measure for traditional computation paradigms. It can be analysed as a weighted time complexity [15], but it can also be derived from the IO complexity [18].

The advantages of neuromorphic computing are primarily reflected by the relatively low energy consumption in comparison with von Neumann architectures. This has motivated the introduction of energy as a new complexity measure, next to time and space complexity [12]. Whereas the energy complexity of a traditional system is usually directly related to its time and space complexity, this is not per se the case for neuromorphic systems. Depending on the type of encoding (voltage, rate or temporal), the spiking behaviour of an SNN allows for sparser information representation. Assuming a binary encoding, the time between two spikes can be interpreted as a number, which only requires energy when the neurons fire. This means that the size of the number (voltage) does not impact the energy complexity of the representation in such an SNN.

The energy complexity in spiking neural networks is measured by the number of spikes, which assumes that they are discrete events of the same value, independent of actual spike amplitude [12]. Under the assumption that spikes are discrete singular events that can happen once per time step, $energy \leq time \cdot space$. Because we do not use a fully spike-based algorithm, but also inspect voltages at times to output and programmed neurons, the assumption that every spike has an energy complexity of $\mathcal{O}(1)$, does not hold. Therefore, we analyse energy both in terms of discrete spikes, and the synaptic currents to give a more exact picture of the energy usage. Both measures of energy are experimentally measured, while the synaptic current is also theoretically computed in terms of the size of the input.

Energy Complexity of the Spiking Neural Network The creation of the SNNs costs energy, given that the SNN consists of neurons with a non-zero initial voltage. This initialisation energy has a complexity of $\mathcal{O}(k^2 \cdot n^2)$. The biggest contributor here is the creation of the network of the update function, wherein n neurons are created that check the $\tilde{\delta}(v_i)$, with initial voltage bound by n. The dependency on k is achieved since the update function initialises a network and is called k^2 times.

The execution of the SNNs has an energy complexity of $\mathcal{O}(k^2 \cdot n^2)$. This is based on the notion that in a fully connected graph, spikes can travel from all neurons to all other neurons, resulting in an energy complexity of $\mathcal{O}(n^2)$. As the LP_{MDS}-approximation performs the update function inside two nested for-loops, the complexity of this algorithm is increased by a factor $\mathcal{O}(k^2)$.

The energy complexity in terms of spikes in the networks is dependent on $\mathcal{O}(k^2 \cdot n + n^2)$. This stems from the fact that we have n spiking neurons in the LP_{MDS}-approximation, but also n spiking neurons in the function in which *Delta* is calculated, with time complexity $\mathcal{O}(n)$.

For the non-spiking functions, we use their time complexity as an approximation method for their energy complexity, where we assume that at each time-step only one computation takes place and all computations cost equal amounts of energy. Under that assumption, the complexity of the complete program, including SNN generation, execution and any necessary non-spiking functions, remains $\mathcal{O}(k^2 \cdot n^2)$.

Energy Complexity of the Kuhn-Wattenhofer Algorithm The energy analysis for the energy used by the Kuhn-Wattenhofer algorithms, for which we again assume that time complexity is a bound for the energy consumption, yields $\mathcal{O}(n)$ and $\mathcal{O}(k^2 \cdot n)$ respectively for LP-relaxation and the LP_{MDS}-approximation. For the energy complexity of the messages, we have used the same assumption, yielding a complexity of $\mathcal{O}(k^2)$.

5 Discussion

We have shown that the distributed algorithm for finding an approximation of the minimum dominating set as presented by Kuhn and Wattenhofer [11] can be successfully implemented in a programmed spiking neural network. This work serves as an example for the porting of distributed algorithms to spiking neural networks and provides subnetworks that can be modularly used in other algorithms.

Complexity analysis shows that the SNN implementation fares worse in terms of time and energy complexity. However, regarding space complexity, the SNN implementation compares favourably to Kuhn and Wattenhofer. The time and energy costs of the initialisation of the spiking neural networks is largely responsible for these seemingly contradicting findings. It is to be noted that Kuhn and Wattenhofer do not take the initialisation of the message-passing system into account. Including the complexity induced by the initialisation in the time

complexity of Kuhn and Wattenhofer, results in a complexity of $\mathcal{O}(k^2 \cdot n + n^2)$. The theoretical time complexity of the algorithm is thus lower compared to the time complexity of the SNN implementation of $\mathcal{O}(k^2 \cdot n^2)$.

Making use of the inherent distributiveness of neural networks may contribute to the field of distributed computing, as the network can be seen as a population of computational units within one device. Neuromorphic architectures may in the future be used in distributed computing applications such as wireless (sensor) networks.

Future research may look into reducing the complexity further to render the time-, space- and energy complexities of the presented SNN implementation on par with the Kuhn-Wattenhofer algorithm. This may be achieved by making full use of the inherent properties of neuromorphic architectures. Another avenue is to efficiently integrate all subnetworks (functions) into one connected network. While the modularity of our implementation is advantageous in that its modules can easily be reused in various kinds of graphs algorithms, particular problems may benefit from one well-tailored network that is not divisible into modules. Lastly, the spiking neural network may be run on neuromorphic hardware

6 Conclusion

This work presents a novel neuromorphic implementation of the distributed minimum dominating set approximation algorithm by Kuhn and Wattenhofer. By programming the network and utilising voltage-based computation within neurons, the LP-relaxation and LP_{MDS}-approximation algorithms as presented by Kuhn and Wattenhofer have been successfully reproduced. The spiking neural networks are simulated in the SNN Simulator [19]. Several spiking neural networks that have been developed in the translation process can function as building blocks for spiking neural network implementations of other (graph) algorithms.

Measuring the time, space and energy complexity of the spiking implementation, we find that it is comparable to the original algorithm. However, the initialisation of the network takes up significant time and energy. As the complexity of the original Kuhn-Wattenhofer algorithm does not take the initialisation of the message-passing structure into account, we conclude that the spiking implementation does not fare significantly worse.

In conclusion, this work demonstrates that programming a spiking neural network is an avenue worth pursuing for scientific computing applications. Furthermore, it shows how leveraging neural properties in the domain of designing spiking implementations of graph problems, prospers on novel insights. Therefore, our work contributes to the scientific body of knowledge of neuromorphic implementations in the field of distributed computing.

References

1. Aimone, J.B., Parekh, O., Phillips, C.A., Pinar, A., Severa, W., Xu, H.: Dynamic programming with spiking neural computing. In: Proceedings of the International

Conference on Neuromorphic Systems. pp. 1–9 (2019)
2. Ali, A., Kwisthout, J.: A spiking neural algorithm for the network flow problem (2019)
3. Bosch, V., Diehl, A., Smits, D., Toeter, A.: SNN implementation of dominating set approximation. https://github.com/a-t-0/spiking-neural-network-of-dominating-set-approximation (2021). https://doi.org/10.5281/zenodo.5496091
4. Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G.A.F., Joshi, P., Plank, P., Risbud, S.R.: Advancing neuromorphic computing with loihi: A survey of results and outlook. Proceedings of the IEEE **109**(5), 911–934 (2021). https://doi.org/10.1109/JPROC.2021.3067593
5. Garey, M.R., Johnson, D.S.: Computers and intractability, vol. 174. freeman San Francisco (1979)
6. Intel: Lava: A software framework for neuromorphic computing. https://github.com/lava-nc, https://github.com/lava-nc
7. Kann, V.: On the approximability of NP-complete optimization problems. Ph.D. thesis, Citeseer (1992)
8. Karbasi, A.H., Atani, R.E.: Application of dominating sets in wireless sensor networks. Int. J. Secur. Its Appl **7**, 185–202 (2013)
9. Kay, B., Date, P., Schuman, C.: Neuromorphic graph algorithms: Extracting longest shortest paths and minimum spanning trees. In: Proceedings of the Neuro-Inspired Computational Elements Workshop. NICE '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3381755.3381762, https://doi.org/10.1145/3381755.3381762
10. Kshemkalyani, A.D., Singhal, M.: Distributed computing: principles, algorithms, and systems. Cambridge University Press (2011)
11. Kuhn, F., Wattenhofer, R.: Constant-time distributed dominating set approximation. Distributed Computing **17**(4), 303–310 (2005)
12. Kwisthout, J., Donselaar, N.: On the computational power and complexity of spiking neural networks. In: Proceedings of the Neuro-Inspired Computational Elements Workshop. NICE '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3381755.3381760, https://doi.org/10.1145/3381755.3381760
13. Li, J., Potru, R., Shahrokhi, F.: A performance study of some approximation algorithms for computing a small dominating set in a graph. Algorithms **13**(12), 339 (2020)
14. Martí, D., Rigotti, M., Seok, M., Fusi, S.: Energy-efficient neuromorphic classifiers. Neural Computation **28**(10), 2011–2044 (10 2016). https://doi.org/10.1162/neco_a_00882, https://doi.org/10.1162/NECO_a_00882, doi: 10.1162/NECO_a_00882
15. Roy, S., Rudra, A., Verma, A.: An energy complexity model for algorithms. In: Proceedings of the 4th Conference on Innovations in Theoretical Computer Science. p. 283–304. ITCS '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2422436.2422470, https://doi.org/10.1145/2422436.2422470
16. Severa, W., Parekh, O., Carlson, K.D., James, C.D., Aimone, J.B.: Spiking network algorithms for scientific computing. In: 2016 IEEE international conference on rebooting computing (ICRC). pp. 1–8. IEEE (2016)
17. Sipser, M.: Introduction to the Theory of Computation. Cengage Learning (01 2012), https://books.google.nl/books?id=1aMKAAAAQBAJ

18. Tran, V.N., Ha, P.H.: Ice: A general and validated energy complexity model for multithreaded algorithms. In: 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS). pp. 1041–1048 (2016). https://doi.org/10.1109/icpads.2016.0138
19. University, R.: Radboud SNN simulator. https://gitlab.socsci.ru.nl/snnsimulator/simsnn, https://gitlab.socsci.ru.nl/snnsimulator/simsnn
20. Xu, Y.Z., Zhou, H.J.: Generalized minimum dominating set and application in automatic text summarization. In: Journal of Physics: Conference Series. vol. 699, p. 012014. IOP Publishing (2016)
21. Yakopcic, C., Rahman, N., Atahary, T., Taha, T.M., Douglass, S.: Solving constraint satisfaction problems using the loihi spiking neuromorphic processor. In: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 1079–1084 (2020). https://doi.org/10.23919/DATE48585.2020.9116227

Refining Weakly-Supervised Free Space Estimation through Data Augmentation and Recursive Training ⋆

François Robinet and Raphaël Frank

Interdisciplinary Centre for Security, Reliability and Trust (SnT)
University of Luxembourg
firstname.lastname@uni.lu

Abstract. Free space estimation is an important problem for autonomous robot navigation. Traditional camera-based approaches rely on pixel-wise ground truth annotations to train a segmentation model. To cover the wide variety of environments and lighting conditions encountered on roads, training supervised models requires large datasets. This makes the annotation cost prohibitively high. In this work, we propose a novel approach for obtaining free space estimates from images taken with a single road-facing camera. We rely on a technique that generates weak free space labels without any supervision, which are then used as ground truth to train a segmentation model for free space estimation. We study the impact of different data augmentation techniques on the performances of free space predictions, and propose to use a recursive training strategy. Our results are benchmarked using the Cityscapes dataset and improve over comparable published work across all evaluation metrics. Our best model reaches 83.64% IoU ($+2.3\%$), 91.75% Precision ($+2.4\%$) and 91.29% Recall ($+0.4\%$). These results correspond to 88.8% of the IoU, 94.3% of the Precision and 93.1% of the Recall obtained by an equivalent fully-supervised baseline, while using no ground truth annotation. Our code and models are freely available online.

Keywords: Weak supervision · Free space estimation · Data augmentation · Recursive training

⋆ This work is supported by the Fonds National de la Recherche, Luxembourg (MASSIVE Project). The authors also thank Foyer Assurances Luxembourg for their support.

1 Introduction

Perception is the first step towards autonomous robot navigation. To be able to safely act in the world, a robot needs to perceive its environment and identify traversable free space. In the context of autonomous driving, free space is usually defined as road areas that are not occupied by either static objects such as traffic signs and road dividers, or by dynamic entities such pedestrians and cars [18]. Since collision-free planning requires a fine-grained understanding of the environment around the vehicle, we attempt to label each pixel of a front-facing camera as traversable or not.

This work focuses on systems that use a single road-facing camera. Monocular free space segmentation has traditionally been approached using supervised segmentation techniques. Although effective, these techniques require vast amounts of pixel-wise annotated frames. Studies have shown that such pixel-level ground truth is significantly more expensive to craft than image-level labels or bounding boxes [27]. In addition to the large labor costs entailed by labeling each frame [7], such approaches are held back by the wide variety of environments and lighting conditions that are present at runtime and need to be captured in training data. This need for ever larger annotated datasets makes supervised learning unsuitable for solving this problem. Instead, we tackle it in a different way: relying on a method that generates weak, noisy, free space annotations without any supervision [42], we train a neural network to generalize past the label noise using data augmentation and recursive training.

Our contributions can be summarized as follows: (1) we study the impact of data augmentation on weakly-supervised free space segmentation, (2) we propose a recursive training scheme that uses a progressively refined ground truth, (3) we establish a new state-of-the-art for weakly supervised free space estimation on the Cityscapes dataset, improving over previous efforts by $+2.3\%$ in IoU, $+2.4\%$ in Precision, and 0.4% in Recall, (4) we discuss the limitations of our simple recursive training approach, and (5) we release our code and models for reproduction and further work.

The remainder of this paper is organized as follows: In Section 2, we review the recent literature for free space estimation, data augmentation in the context of semantic segmentation, and recursive training. In Section 3, we introduce our data augmentation and recursive training schemes. In Section 4, we describe our use of the Cityscapes dataset [7] and detail the experimental setup of this study. In Section 5, we carry out experiments and present the qualitative and quantitative results achieved. Finally, we summarize our contributions and share further research directions.

2 Related Work

Over the last decades, free space estimation has been approached with methods that leverage a wide variety of sensors, *e.g.* GNSS [24], LiDAR [45] or cameras [35]. In this work, we place a particular focus on recent camera-based learning methods that use Convolutional Neural Networks (CNNs). Our work builds on recent advances in network architectures for segmentation and on unsupervised methods specific to free space estimation. We present this background material in the following sections.

2.1 Supervised Learning for Segmentation

As a segmentation task, supervised free space estimation has directly benefited from progress in semantic segmentation. Pixel-level prediction carries a crucial challenge for network design: an optimal prediction can only be achieved by combining fine-grained local information with global contextual cues. Fully Convolutional Networks (FCNs) rely on skip connections to carry these cues in their encoder-decoder architecture [28], while SegNets ease the upsampling task by reusing encoder max-pooling indices in the decoder [3]. Building on similar ideas, U-Nets combine entire encoder feature maps with decoder features at each step of the expansion path of the network [40]. U-Nets have attracted a lot of attention in recent years, and researchers have proposed refinements such as the use of dense connections [19] and dilated convolutions [51], the integration of attention mechanisms [34], or extensions to volumetric images [32]. In this work, we will rely on a simple U-Net architecture. Our choice is motivated by a recent finding that many recent architecture improvements are outperformed by a well-tuned vanilla U-Net [17].

2.2 Weakly-Supervised Semantic Segmentation

The major drawback of supervised techniques is their reliance on extensive human-annotated datasets. The cost of labeling is particularly important in segmentation tasks, where the total time required to annotate every pixel in a single frame can reach 1.5 hours in some cases [7]. The reuse of models pre-trained on very large datasets such as ImageNet [11] partially alleviates this problem, but several thousands of training images are still routinely needed to reach adequate performance. In recent years, researchers have devised strategies to reduce or eliminate the need for human annotations during training.

In cases where fine-grained annotations are available for at least a subset of the data, semi-supervised approaches such as Co-Training can be applied [37]. In the complete absence of pixel-wise ground truth labels, researchers have proposed to use domain adaptation from synthetic datasets [16], or to rely on weaker ground truth. Existing techniques rely on coarser labels, such as bounding boxes [9,20,21,46], image-level labels [38,12,43], class activation maps [5], single points [4], or scribbles [26].

2.3 Unsupervised and Weakly-Supervised Monocular Free Space Segmentation

Monocular free space estimation has been approached in many different ways that differ in the representation they use. Stixel-like approaches represent obstacles as verti-

cal sticks [2,8] or horizontal curves [48], but ignore free space lying behind obstacles. Monocular SLAM relies on video sequences to obtain point-clouds which do not explicitly represent free space [13,33,10]. Using temporal sequences and structure-from-motion to jointly learn an explicit representation of free space and obstacle footprints has also been recently proposed [44]. Our work uses a different strategy: we learn dense free space estimates from single frames using approximate masks that are obtained without human-supervision. Such *weak labels* have historically been generated using depth information from stereo pairs before localizing the ground plane, for example using the v-Disparity algorithm [23,14,31]. Other attempts exploit strong road texture and location priors, by dividing the input into superpixels and clustering them based on saliency maps [43] or semantic features [35]. We stress that using weak labels departs from previously mentioned approaches that leverage coarse ground truth, since weak labels contain false positives and negatives.

2.4 Training Strategies for Weakly-Supervised Segmentation

Recent research shows that it is possible to train over-parametrized models to generalize past some of the label noise using Stochastic Gradient Descent (SGD) schemes combined with early stopping [25]. Dealing with label noise at training time has become an important research area over the past few years. Solutions to this problem include label cleaning [6], noise-aware network architectures [41], or noise reduction through robust loss functions [30,29,39].

Besides work on training algorithms themselves, researchers have also largely explored regularization through data augmentation in unsupervised settings. Traditional augmentation strategies (scaling, color jittering, flipping, cropping, *etc.*) change pixel values in a single input image without altering its semantic content. More recently, researchers have proposed augmentations that combine several images and their labels. Two notable examples are MixUp [50] and CutMix [49]. MixUp is a method that augments the training set using convex combinations of image pairs and labels, while CutMix overlays random crops of other samples on top of original frames.

3 Methodology

In this work, we train U-Net models to predict dense free space from RGB images by learning on approximate labels that can be generated without any supervision. Since our focus is on improving training aspects rather than on improving weak labels generation, we will reuse the weak labels from [42]. We look at improving training across two dimensions: data augmentation and recursive training.

3.1 Data Augmentation

We study the impact of data augmentation on weakly-supervised free space estimation. We cover both traditional augmentation techniques that operate on single images, as well as MixUp and CutMix, which are more recent and combine multiple samples.

Color-Flip-Crop To represent traditional augmentation techniques, we use a combination of color jittering, horizontal flips and random cropping, which we will refer to as *Color-Flip-Crop* or *CFC* in the remainder of the text. Each augmentation is independently applied with a 50% probability. The color jittering randomly affects brightness, contrast, saturation, and hue using the bounds defined in the Torchvision implementation [1]. In order to preserve most of the original image, cropping is performed with a randomly chosen rectangle that occupies between 25% and 50% of the image area. The aspect ratio is also randomly chosen, with the constraint that the height is at least 10% of the height of the original image. Figure 1 shows some examples of the effect of CFC on a single randomly chosen training image.

Fig. 1: Seven possible Color-Flip-Crop augmentations on a random training sample. The original sample is on the top-left. We show ground truth mask for illustration purposes, they are not used during training.

MixUp Rather than augmenting isolated images, Mixup trains models on convex combinations of samples [50]. By training on synthesized samples that lie between the original training samples, MixUp encourages the network to exhibit a linear behavior between samples and helps preventing memorization. During training, each sample (x_1, y_1) is combined with another random sample (x_2, y_2) from the batch using Equations 1 and 2, where we sample λ uniformly in $[0, 1]$. The effect of combining input samples is illustrated on Figure 2.

$$x_{mixup} = \lambda x_1 + (1 - \lambda) x_2 \tag{1}$$

$$y_{mixup} = \lambda y_1 + (1 - \lambda) y_2 \tag{2}$$

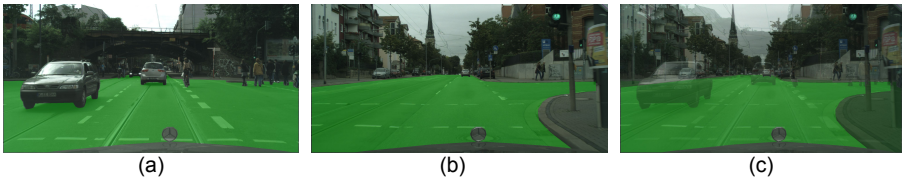


Fig. 2: MixUp augmentation combining two random samples (a) and (b) from the training set. The convex combination using $\lambda = 0.5$ is shown as (c). We show ground truth mask for illustration purposes, they are not used during training.

CutMix Similar to Mixup in spirit, CutMix also combines two random input samples (x_1, y_1) and (x_2, y_2) from the same batch [49]. Rather than combining them over the entire image, CutMix overlays a crop of x_2 over x_1, and the same crop of y_2 over y_1. Equations 3 and 4 formalize this process using a random binary mask $M \in \{0, 1\}^{H \times W}$ to denote the cropped area ($\circ$ denotes the element-wise product). Like for the CFC augmentation, the cropping mask M occupies between 25% and 50% of the image area with a random aspect ratio. Figure 3 illustrates four different instances of CutMix augmentation on a chosen training sample. CutMix generates more natural images than MixUp and allows the network to learn more localizable features since the transformation is only applied to a fraction of the input image.

$$x_{cutmix} = (1 - M) \circ x_1 + M \circ x_2 \tag{3}$$

$$y_{cutmix} = (1 - M) \circ y_1 + M \circ y_2 \tag{4}$$

3.2 Recursive Training

We are training neural networks to estimate free space by learning on approximate labels y_{weak}. Since neural networks trained with SGD variants are partially robust to noise in their training targets [25], the outputs y will tend to approximate the unknown ground truth y^* better than y_{weak}. Assuming the outputs y are better estimates of free space than y_{weak}, it is natural to treat them as cleaner targets for a second round of training. This process can in principle be iterated to obtain progressively cleaner outputs y_2, y_3, *etc*. This approach was already attempted in the context weakly-supervision free space segmentation [43], but we revisit its impact in the presence of data augmentation and with different weak labels. Figure 4 illustrates the process for a given training round.

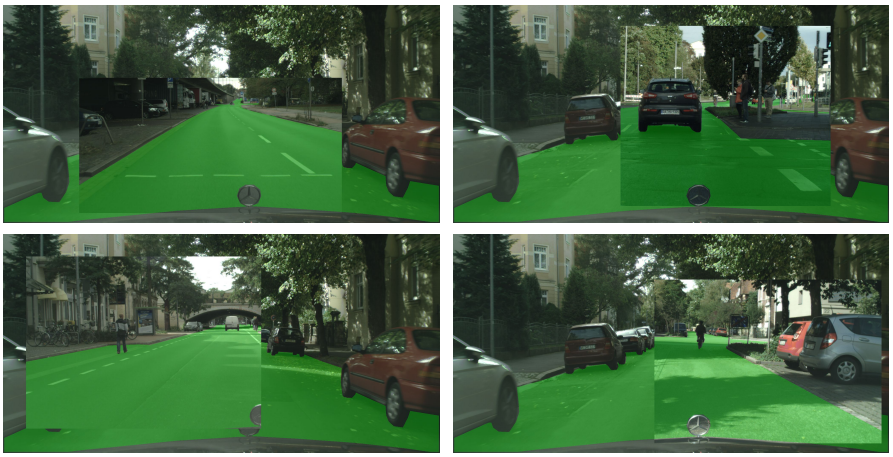


Fig. 3: Four instances of the CutMix augmentation on a random training sample.
We show ground truth mask for illustration purposes, they are not used during training.

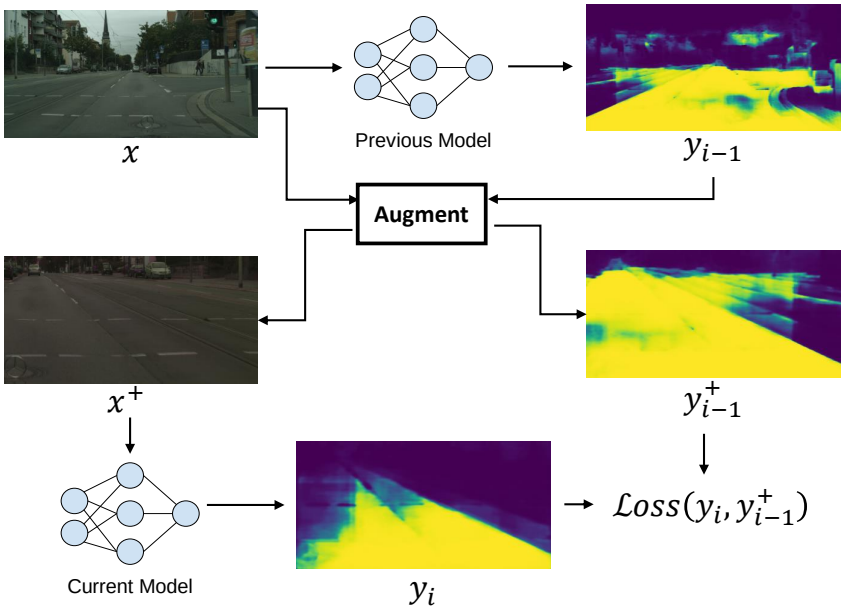


Fig. 4: Recursive training procedure. The current model is trained on augmented outputs from the model obtained at the previous training round. In this example, CFC is used for augmentation. The process is similar for other augmentation strategies.

4 Experimental Setup

4.1 Dataset

Our experiments leverage the Cityscapes dataset, which provides pixel-wise ground truth labels for 30 visual classes in 5000 frames [7]. The official test set has no public annotation, and we therefore treat the 500 frames of its validation set as our test set and randomly split the Cityscapes training set into 2380 training and 595 validation frames. Since we are interested in estimating drivable free space in the context of autonomous vehicle navigation, we consider free space equivalent to the *road* class. Cityscapes also contains 1.6% of frames with no *road* pixel. For these frames, visual inspection confirmed that free space correspond to the *ground* class, and that label was used for free space instead of *road*. Finally, the semantic labels include 6 *void* classes such as *unlabeled*, *out of the region of interest* or *ego-vehicle*. Following official Cityscapes segmentation benchmarks, we ignore pixels corresponding to such classes at evaluation time using a binary evaluation mask. We note that this evaluation mask is never used during training or validation, only to evaluate models on the test set.

4.2 Evaluation Metrics

We use three evaluation metrics: the Intersection-over-Union (IoU), Precision and Recall. IoU is a standard metric in segmentation tasks to reflect the overall quality of the predictions. However, IoU does not immediately capture *false free space positives*. These pixels that are labeled as part of the road but are actually occupied are extremely harmful in robotic path-planning scenarios. For this reason, we also monitor the Precision of the free space class, *i.e.* the fraction of our free space prediction that is indeed free space. To obtain a complete picture of prediction quality, we also monitor Recall. We however note that missing free space in predictions has less impact than false free space positives in robot navigation contexts. Given a single free space prediction $\hat{y}$, ground truth y, and evaluation mask m, the metrics for a single frame of shape $H \times W$ are computed with Equations 5, 6 and 7, where $\hat{y},\ y,\ m \in \{0,1\}^{H \times W}$.

$$IoU = \frac{\sum_i \hat{y}_i y_i m_i}{\sum_i (\hat{y}_i + y_i - \hat{y}_i y_i) m_i} \tag{5}$$

$$Precision = \frac{\sum_i \hat{y}_i y_i m_i}{\sum_i \hat{y}_i m_i} \tag{6}$$

$$Recall = \frac{\sum_i \hat{y}_i y_i m_i}{\sum_i y_i m_i} \tag{7}$$

4.3 Network architectures

Following recent research that shows that a well-tuned vanilla U-Net can outperform many refined variants on most segmentation tasks [17], we opt for a U-Net structure based on a ResNet18 residual network backbone [40,15,47]. To allow for comparison with prior art, we also implement and train the SegNet model described in [42]. For

computational reasons, we use a 512×1024 input resolution in all experiments. Outputs are however re-scaled using nearest neighbor interpolation in order to compute IoU and Precision in the original 1024×2048 resolution.

4.4 Training procedure

We use the PyTorch framework [36] and train randomly initialized models to minimize a binary cross-entropy loss using the Adam optimizer [22], a batch size of 8 and an initial learning rate of 0.001. We train our models on single NVIDIA V100 for up to 200 epochs, with an early stopping strategy that halts training when the validation loss has not improved by at least 10^{-4} for 50 consecutive epochs. For each experiment, we select the model that minimizes the validation loss.

4.5 Use of ground truth data

The Cityscapes dataset provides ground truth annotations for all training and validation frames used in this study. We stress that these annotations are only used to train the fully-supervised baseline for comparison with our weakly-supervised approach. Outside of the fully-supervised experiment, ground truth labels are never used for training, hyperparameter tuning, or to perform early stopping. Ground truth IoU, Precision and Recall are computed only once on the test set, after all these steps have been performed.

5 Results

This section describes the experiments carried out to benchmark our proposed method, using Precision, IoU and Recall. We present results for three main categories of models: 1) a fully-supervised upper-bound, 2) unsupervised and weakly-supervised baselines, and 3) U-Nets trained on the weak labels using recursive training and different augmentation strategies. The quantitative results for each category are summarized in Table 1. In this section, we analyze the results of each category, discuss the limitations of recursive training, and present qualitative results.

5.1 Fully-Supervised Results

Since Cityscapes provides pixel-wise ground truth annotations for our training and validation data, we use it to train a fully-supervised U-Net for comparison with its unsupervised counterpart. When trained on ground-truth labels, our U-Net model reaches high IoU (94.12%), Precision (97.26%) and Recall (97.27%). Since this fully-supervised model is the only one that uses ground truth labels at any point during training and validation, it is expected to produce an upper-bound for our unsupervised experiments.

5.2 Unsupervised and Weakly-Supervised Baselines

Competing unsupervised approaches are often focused on generic semantic segmentation rather than free space estimation, and use other datasets than Cityscapes as benchmarks [9,46,38,12,5]. Among weakly-supervised approaches that tackle free space estimation [14,48,43,16], only two publish results for Cityscapes. *Distant Supervision* [43] and *Unsupervised Domain Adaptation* [16] respectively obtain an IoU of 80% and 70.4%, but do not report Precision or Recall values.

We generate approximate labels without supervision using the technique described in [42]. Evaluating these raw weak labels, we obtain an IoU of 79%, a Precision of 87.78% and a Recall of 89.24%. These results can be further improved by training a neural network to generalize beyond the noise in these labels. This was already attempted using the SegNet architecture in [42], which we also implement and train for comparison. SegNet is able to improve results over raw weak labels in IoU ($+2.3\%$), Precision ($+1.58\%$) and Recall ($+0.91\%$).

5.3 Data Augmentation & Recursive Training

We train the same U-Net model using different data augmentation strategies. Since the outputs of our different augmented U-Nets are better than the initial weak labels, we use them as target for a second round of training. We iterate this recursive training process four times for each of the data augmentation strategies under study. We limit training to four rounds for computational reasons and because it is enough for IoU values to reach their peak.

No Augmentation We start by training a U-Net with the weak labels as targets and without any data augmentation. We observe that it compares favorably with the results from SegNet, reaching an IoU of 81.85%, a Precision of 90.65%, and a Recall of 89.76%. Without resorting to data augmentation, recursive training over several rounds is unable to meaningfully improve IoU, and slightly decreases Precision in favor of Recall.

MixUp Applying MixUp allows to improve Precision compared to not using data augmentation by 0.5% in the first training round. IoU is maintained, but Recall decreases by 0.45%. Iterative training is however not effective when combined with MixUp, since we observe a drop in Precision after each round. As discussed in Section 4.2, free space IoU and Precision are more important than Recall in an autonomous navigation scenario. In this case, increases in Recall are not enough to compensate this effect, and we observe a steady decrease in IoU.

Color-Flip-Crop Traditional data augmentation consisting of color jittering, horizontal flips and random cropping is able to improve IoU over not using augmentation and over using MixUp. After a single training round, CFC allows to reach an IoU of 81.99% through increasing Recall by 1.47% compared to the first round without augmentation. Subsequent training rounds are able to improve both Precision and IoU. After 3 iterations, the model reaches an IoU of 82.34% and a Precision of 90.75%.

CutMix The CutMix augmentation can be seen as providing the advantages of cropping and MixUp. Like MixUp, it synthesizes new input samples by combining pairs of existing ones. However, CutMix produces more natural images and its effect is localized since it only affects the area of a random crop. The locality of CutMix has been shown to allow models to learn more localizable features in classification scenarios [49], and it is not surprising that such features are helpful in this segmentation context. Indeed, models trained with CutMix augmentation outperform all other models by a wide margin. After a single training round, CutMix improves over not using augmentations in IoU ($+1.2\%$), Precision ($+0.5\%$), and Recall ($+0.26\%$).

Since our application scenario favors Precision over Recall, our best overall model is obtained after the fourth training round, reaching an IoU of 83.64% and a Precision of 91.75%. Compared to the prior state-of-the-art results from SegNet [42], it improves IoU by 2.3%, Precision by 2.4% and Recall by 0.4%. Although our model does not rely on any human-annotated ground truth, its relative performance compared to the fully-supervised variant is impressive: we reach 88.8% of its IoU, 94.3% of its Precision, and 93.1% of its Recall.

5.4 Limits of Recursive Training

While CutMix results are impressive, we note that the success of recursive training is limited. When not applying data augmentation or when using MixUp, recursive training does not improve on IoU or Precision. In the case of CFC and CutMix augmentations, results are more encouraging, but the improvements are limited to three rounds of training. Starting with the fourth round of training, IoU results start to degrade, sometimes getting worse than those obtained after a single round of training. Explaining

this effect is not straightforward: given that target labels on round 4 are superior to those used on round 3 in both IoU and Precision, we would expect to either observe improved or plateauing results. Such recursive training strategy has been successfully used in foreground class segmentation contexts with results improving over more than 10 rounds [21]. As opposed to our completely unsupervised approach, the authors of [21] could exploit coarser ground truth in the form of bounding boxes in order to refine predictions after each round. We postulate that the absence of such refinement step in our approach is the reason we are unable to further leverage recursive training. Designing such a prediction refinement step will be the topic of future work.

	Training/Validation Labels	**Test IoU**	**Test Precision**	**Test Recall**
Fully-Supervised U-Net	ground truth	94.12%	97.26%	97.27%
Unsup. Domain Adaptation [16]	synthetic data	70.40%	not reported	not reported
Distant Supervision [43]	image labels	80.00%	not reported	not reported
Weak Labels [42]	no training	79.00%	87.78%	89.24%
SegNet (repr. from [42])	weak labels	81.30%	89.36%	90.15%
U-Net (no augmentation)				
Round 1	weak labels	81.85%	90.65%	89.76%
Round 2	output of round 1	81.79%	89.53%	90.80%
Round 3	output of round 2	81.86%	90.15%	90.27%
Round 4	output of round 3	81.82%	90.11%	90.25%
U-Net + MixUp				
Round 1	weak labels	81.89%	91.14%	89.31%
Round 2	output of round 1	81.97%	90.89%	89.60%
Round 3	output of round 2	81.62%	90.13%	89.97%
Round 4	output of round 3	81.45%	89.91%	90.02%
U-Net + Color-Flip-Crop				
Round 1	weak labels	81.99%	88.80%	91.23%
Round 2	output of round 1	82.12%	89.71%	90.64%
Round 3	output of round 2	82.34%	90.75%	90.69%
Round 4	output of round 3	81.91%	90.21%	90.27%
U-Net + CutMix				
Round 1	weak labels	83.05%	91.19%	90.51%
Round 2	output of round 1	83.58%	91.20%	91.12%
Round 3	output of round 2	**83.77%**	91.23%	**91.29%**
Round 4	output of round 3	83.64%	**91.75%**	90.62%

Table 1: Results on the Cityscapes validation set, which we treat as our test set. The best results for a given data augmentation strategy are underlined, and the best overall results are reported in bold.

5.5 Qualitative Results

We compare the free space estimates from weak labels with the predictions of our best model on test set samples on Figure 5.

The ability of our learned model to generalize past some of the noise present in the weak labels that were used during training is clearly visible in the first two rows of Figure 5. Indeed, the cars and side walks that were wrongly considered free space in the weak labels are correctly predicted by our trained model. In addition to its higher Precision, our model also has higher IoU and Recall, as illustrated by the near-absence of orange areas in its predictions.

The third row shows a more contrasted situation. Although our model is able to cover more free space, it still shows some signs of overfitting to noise in the weak labels. Shadows are especially problematic because they are likely to impact the superpixel segmentation that the weak labels are based on, resulting in missed free space areas such as the one present in front of the cyclist. Since this effect happens fairly consistently over the training set, our model is incapable of completely addressing it.

Finally, the fourth row illustrates another partial failure of our model in a particularly crowded scene. Compared to the corresponding weak labels, the trained model correctly rejects pedestrians, but is unable to produce a clean segmentation around them and considers the pavement as occupied space. Although the prediction still contains errors, we note that red areas in our prediction are much more acceptable from a semantics point-of-view than the ones from the corresponding weak labels.

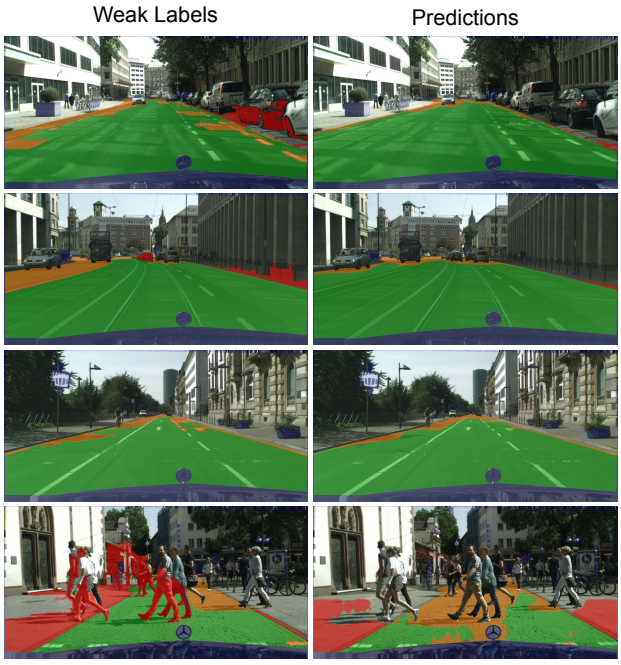


Fig. 5: Qualitative results from the test set obtained from a U-Net trained with CutMix for 4 rounds. Predictions are color-coded using the ground truth: green and red respectively corresponds to correct and incorrect predictions, orange represents missing free space, and areas that are ignored at evaluation time are denoted in blue (see Section 4.1).

6 Conclusion

In this work, we investigate different weakly-supervised training strategies for teaching a neural network to predict free space from images taken with a single road-facing camera. Our models are trained using weak labels that are generated without human intervention, and we investigate the impact of recursive training with several data augmentation schemes. We show that the CutMix augmentation is particularly efficient for free space estimation, especially when combined with recursive training. We benchmark our results on the Cityscapes dataset and improve over unsupervised and weakly-supervised baselines, reaching 83.64% IoU ($+2.3\%$), 91.75% Precision ($+2.4\%$) and 91.29% Recall ($+0.4\%$). Our best model obtains 88.8% of the IoU, 94.3% of the Precision and 93.1% of the Recall of the fully-supervised competitor that trains from expensive pixel-wise labels. Finally, we show that simple recursive training is limited in its ability to increase performances, and suggest directions to improve the approach. Future work will also investigate improvements to weak label generation and applications to more general segmentation scenarios.

References

1. Torchvision: Datasets, transforms and models specific to computer vision. https://github.com/pytorch/vision (2021)
2. Badino, H., Franke, U., Pfeiffer, D.: The stixel world - a compact medium level representation of the 3d-world. In: Denzler, J., Notni, G., Süße, H. (eds.) Pattern Recognition. pp. 51–60. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)
4. Bearman, A., Russakovsky, O., Ferrari, V., Fei-Fei, L.: What's the point: Semantic segmentation with point supervision. In: Computer Vision – ECCV 2016. pp. 549–565. Lecture Notes in Computer Science (LNCS), Springer International Publishing (Sep 2016). https://doi.org/10.1007/978-3-319-46478-7_34, http://www.eccv2016.org/, 14th European Conference on Computer Vision 2016, ECCV 2016 ; Conference date: 08-10-2016 Through 16-10-2016
5. Chang, Y., Wang, Q., Hung, W., Piramuthu, R., Tsai, Y., Yang, M.: Mixup-cam: Weakly-supervised semantic segmentation via uncertainty regularization. In: 31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020. BMVA Press (2020), https://www.bmvc2020-conference.com/assets/papers/0367.pdf
6. Chiaroni, F., Rahal, M.C., Hueber, N., Dufaux, F.: Hallucinating a Cleanly Labeled Augmented Dataset from a Noisy Labeled Dataset Using GANs. In: IEEE (ed.) 26th IEEE International Conference on Image Processing (ICIP). Taipei, Taiwan (Sep 2019), https://hal.archives-ouvertes.fr/hal-02054836
7. Cordts, M., Omran, M., Ramos, S., Scharwächter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset. In: CVPR Workshop on the Future of Datasets in Vision. vol. 2 (2015)
8. Cordts, M., Rehfeld, T., Schneider, L., Pfeiffer, D., Enzweiler, M., Roth, S., Pollefeys, M., Franke, U.: The stixel world: A medium-level representation of traffic scenes. Image and Vision Computing **68** (02 2017). https://doi.org/10.1016/j.imavis.2017.01.009
9. Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: Proceedings of the IEEE international conference on computer vision. pp. 1635–1643 (2015)
10. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6), 1052–1067 (2007). https://doi.org/10.1109/TPAMI.2007.1049
11. Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
12. Durand, T., Mordan, T., Thome, N., Cord, M.: Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5957–5966 (2017). https://doi.org/10.1109/CVPR.2017.631
13. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 834–849. Springer International Publishing, Cham (2014)
14. Harakeh, A., Asmar, D., Shammas, E.: Identifying good training data for self-supervised free space estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)

15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
16. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. CoRR **abs/1612.02649** (2016), http://arxiv.org/abs/1612.02649
17. Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P.F., Kohl, S., Wasserthal, J., Koehler, G., Norajitra, T., Wirkert, S.J., Maier-Hein, K.H.: nnu-net: Self-adapting framework for u-net-based medical image segmentation. CoRR **abs/1809.10486** (2018), http://arxiv.org/abs/1809.10486
18. Janai, J., Güney, F., Behl, A., Geiger, A.: Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. ArXiv **abs/1704.05519** (2020)
19. Jégou, S., Drozdzal, M., Vázquez, D., Romero, A., Bengio, Y.: The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 1175–1183 (2017)
20. Kervadec, H., Dolz, J., Wang, S., Granger, E., ben Ayed, I.: Bounding boxes for weakly supervised segmentation: Global constraints get close to full supervision. In: Medical Imaging with Deep Learning (2020), https://openreview.net/forum?id=VOQMC3rZtL
21. Khoreva, A., Benenson, R., Hosang, J., Hein, M., Schiele, B.: Simple does it: Weakly supervised instance and semantic segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1665–1674 (2017). https://doi.org/10.1109/CVPR.2017.181
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2015)
23. Labayrade, R., Aubert, D., Tarel, J.P.: Real time obstacle detection in stereovision on non flat road geometry through" v-disparity" representation. In: Intelligent Vehicle Symposium, 2002. IEEE. vol. 2, pp. 646–651. IEEE (2002)
24. Laddha, A., Kocamaz, M.K., Navarro-Serment, L.E., Hebert, M.: Map-supervised road detection. In: 2016 IEEE Intelligent Vehicles Symposium (IV). pp. 118–123 (2016). https://doi.org/10.1109/IVS.2016.7535374
25. Li, M., Soltanolkotabi, M., Oymak, S.: Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In: International Conference on Artificial Intelligence and Statistics. pp. 4313–4324. PMLR (2020)
26. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3159–3167 (2016). https://doi.org/10.1109/CVPR.2016.344
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
28. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
29. Lu, Z., Fu, Z., Xiang, T., Han, P., Wang, L., Gao, X.: Learning from weak and noisy labels for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 486–500 (03 2017). https://doi.org/10.1109/TPAMI.2016.2552172
30. Mairal, J., Elad, M., Sapiro, G.: Sparse representation for color image restoration. Trans. Img. Proc. **17**(1), 53–69 (Jan 2008). https://doi.org/10.1109/TIP.2007.911828, https://doi.org/10.1109/TIP.2007.911828
31. Mayr, J., Unger, C., Tombari, F.: Self-supervised learning of the drivable area for autonomous vehicles. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 362–369. IEEE (2018)

32. Milletari, F., Navab, N., Ahmadi, S.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). pp. 565–571 (2016). https://doi.org/10.1109/3DV.2016.79
33. Newcombe, R., Lovegrove, S., Davison, A.: Dtam: Dense tracking and mapping in real-time. pp. 2320–2327 (11 2011). https://doi.org/10.1109/ICCV.2011.6126513
34. Oktay, O., Schlemper, J., Folgoc, L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N., Kainz, B., Glocker, B., Rueckert, D.: Attention u-net: Learning where to look for the pancreas (04 2018)
35. Oliveira, G.L., Burgard, W., Brox, T.: Efficient deep models for monocular road segmentation. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4885–4891 (2016). https://doi.org/10.1109/IROS.2016.7759717
36. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
37. Peng, J., Estrada, G., Pedersoli, M., Desrosiers, C.: Deep co-training for semi-supervised image segmentation (2019)
38. Pinheiro, P.O., Collobert, R.: From image-level to pixel-level labeling with convolutional networks. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1713–1721 (2015). https://doi.org/10.1109/CVPR.2015.7298780
39. Robinet, F., Demeules, A., Frank, R., Varisteas, G., Hundt, C.: Leveraging privileged information to limit distraction in end-to-end lane following. In: 2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC). pp. 1–6 (2020). https://doi.org/10.1109/CCNC46108.2020.9045110
40. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
41. Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., Fergus, R.: Training convolutional networks with noisy labels (Jan 2015), 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015
42. Tsutsui, S., Kerola, T., Saito, S., Crandall, D.J.: Minimizing supervision for free-space segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 988–997 (2018)
43. Tsutsui, S., Saito, S., Kerola, T.: Distantly supervised road segmentation. 2017 IEEE International Conference on Computer Vision Workshops (ICCVW) pp. 174–181 (2017)
44. Watson, J., Firman, M., Monszpart, A., Brostow, G.J.: Footprints and free space from a single color image. In: Computer Vision and Pattern Recognition (CVPR) (2020)
45. Xiao, L., Dai, B., Liu, D., Hu, T., Wu, T.: Crf based road detection with multi-sensor fusion. In: 2015 IEEE Intelligent Vehicles Symposium (IV). pp. 192–198 (2015). https://doi.org/10.1109/IVS.2015.7225685
46. Xie, W., Wei, Q., Li, Z., Zhang, H.: Learning effectively from noisy supervision for weakly supervised semantic segmentation. In: BMVC (2020)
47. Yakubovskiy, P.: Segmentation models. https://github.com/qubvel/segmentation_models (2019)
48. Yao, J., Ramalingam, S., Taguchi, Y., Miki, Y., Urtasun, R.: Estimating drivable collision-free space from monocular video. In: 2015 IEEE Winter Conference on Applications of Computer Vision. pp. 420–427 (2015). https://doi.org/10.1109/WACV.2015.62

49. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 6022–6031 (2019)
50. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. International Conference on Learning Representations (2018)
51. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6230–6239 (2017). https://doi.org/10.1109/CVPR.2017.660

Object detection with semi-supervised adversarial domain adaptation for real-time edge devices

Mattias Billast, Tom De Schepper, Kevin Mets,
Peter Hellinckx, José Oramas, and Steven Latré

University of Antwerp - imec, IDLab, Department of Computer Science,
Sint-Pietersvliet 7, 2000 Antwerp, Belgium.

Abstract. Object detection on real-time edge devices for new applications with no or a limited amount of annotated labels is difficult. Where traditional data-hungry methods fail, transfer learning can provide a solution by transferring knowledge from a source domain to the target application domain. We explore domain adaptation techniques on a one-stage detection architecture, i.e. YOLOv3, which enables use on edge devices. Existing methods in domain adaptation with deep learning for object detection, use two-stage detectors like Faster-RCNN with adversarial adaptation. By using a one-stage detector, the speed increases by a factor of eight. With our proposed method, we reduce by 28% the changes in performance introduced by the gap between the source and target domains.

Keywords: Domain adaptation · Object detection · adversarial learning.

1 Introduction

Object detection and classification are amongst the main tasks addressed by computer vision [30]. They are used in a wide variety of application domains like autonomous driving, robotics, medical imaging, tracking of various subjects, counting, manufacturing, etc. In general, deep learning techniques are applied which require significant amounts of examples. The performance of deep neural networks with abundantly available labels surpasses other techniques. Examples of such use cases are car detection and classification of written characters [11][15].
Most of these deep neural networks also require a GPU which provides the necessary computing power. Hence, the main obstacles for the adoption of these supervised machine learning approaches in new applications remain the lack of (labeled) data and the needed computing power for inference. This last factor clearly limits their use in edge devices. Often, there already exist application domains with similar properties and labeled data which can be used as a starting point, i.e. the source domain. Ideally, the knowledge from the source domain can be transferred to the new application domain, the target domain. New application domains often do not have enough labels. Examples of these include the detection of animals other than a cat or dog, autonomous vehicles other than cars, and even detecting the same subject in another environment/dataset can cause the source model to have a significant drop in performance. Therefore, an automated framework to adapt a source model to a target domain with only a few target

labels can prevent the time- and cost-consuming task of labelling a large dataset.
Existing methods[9][3] for domain adaptation rely on creating domain-invariance between source and target domain. This can be done by adversarially changing the feature encodings from convolutional layers or creating synthetic images which close the gap between the two domains. These techniques will be discussed in detail in Section 2. Regarding domain adaptation for the object detection task, most efforts from the literature are based on the Faster-RCNN detector [9]. Faster-RCNN is a good choice for applications with sufficient computational power or when no real-time inference is needed.
This is even more critical given the increasing number of new use cases where computations are expected to take place in real-time on-site, instead of on a remote cloud server [31]. With limited resources and/or the need for a real-time application, faster frameworks like one-stage detectors provide opportunities to meet the demand. By using a one-stage network, e.g. YOLOv3 [22], as the backbone network to perform object detection, the use of edge devices in real-time is made possible. This is mainly due to the inference speed advantage of YOLO over Faster-RCNN [23].
A good application example is an autonomous vessel that needs real-time tracking of the vessels in the near distance with the computational power on-board. Maritime vessels scan the whole environment with a radar once or twice every second [2]. This is sufficient due to their low speed. To make the step towards autonomous vessels, a camera and/or LiDar sensor needs to be added to be able to make navigation decisions with a more comprehensive understanding of the environment. If a camera can locate and classify objects in the water with the computation power on-board, in synchronization with the radar, then this could constitute a leap forward for the maritime industry.
Taking the previous application setting into account, in this paper we present a transfer learning technique based on feature adaptation that uses the labeled data in a source domain to improve the performance in a target domain with no or limited labeled data. This effectively increases the overall generalization and robustness of the source model. The performance is compared against other transfer learning techniques like cycleGAN image adaptation [33] and combining feature adaptation with image adaptation. To validate the different transfer learning techniques, two experiments are set up. First, the different techniques are used to transfer knowledge from one dataset to another when detecting the same subject (i.e. cars). The two datasets used in are COCO2017 [18] and KITTI [7]. Second, the techniques are used to detect similar classes from the same dataset, i.e. learning to properly detect a lion by transferring the obtained knowledge from detecting a tiger. The dataset used for this task is OpenImages. In both cases, there are 30 labelled target images available to fine-tune the source model. With feature adaptation, there is a 5 to 9% improvement of the mean Average Precision (mAP) compared to the fine-tuned source model.
To summarize, we propose domain adaptation techniques based on feature alignment and synthetic image alignment with fast real-time object detection models that enable use on the edge with limited labeled data.

2 Related work

The proposed method lies at the intersection of object detection and domain adaptation. As such, we will position our system with respect to efforts addressing those tasks.

2.1 Object detection

To perform object detection, the subject first has to be localised and then classified. There are two main categories for object detection models, i.e. traditional models without deep learning and models with deep learning.

Engineered Features. SIFT [20] detects objects in the image by matching local features which are scale- and orientation-invariant. SURF [1] uses a similar feature descriptor as SIFT but speeds up the process significantly by the integral image for image convolutions and simplifying the overall method. Other feature descriptors such as Haar-like features [16], HOG [5], and ORB [24] perform similarly. They all have their advantages depending on shape, colour, texture, and illumination. On the one hand, these methods have the advantage of being relatively lightweight, they are outperformed by their counterparts based on deep neural networks. This discourage their use in critical applications.

Learning-based Representations. With the advent of big data and increased computational power, representation learning methods got a lot of interest. Moreover, in computer vision, all the current State Of The Art (SOTA) methods are based on deep convolutional neural networks [13]. Their success is attributed to the large number of parameters present in Deep Neural Networks (DNN), which can be used to model all the possible variations of how an object is depicted. Faster-RCNN [23] is a very commonly used two-stage model which uses a Region Proposal Network (RPN). This is based on the feature encodings after multiple convolutional layers to first propose possible bounding boxes to focus on. In the next step, these region proposals are used to locate the best proposals and classify the object. To speed up the whole process, there are one-stage models such as YOLO [21] and SSD [19] without the RPN, making it more a regression/classification model. YOLOv3 [22] improves YOLO with bounding boxes at three different scales by using a similar idea as a Feature Pyramid Network [17] and with increased frames per second (fps). There is a small drop in performance from a two-stage to a one-stage model but the gained speed enables to detect objects in real-time, even with less computational power.

2.2 Domain adaptation

The focus of this paper is to improve object detection performance when operating on setting with no or limited annotated data is available. While there are different options to apply transfer learning, they all involve domain-invariance between source and target domain [32][8][33][29][9]. A possible method is to map extracted features, which are the input to the domain classifier, from source to target domain or the other way around [32]. Another option is to change the style of an image synthetically from source to target domain or vice-versa. This mapping is primarily done by a Generative

Adversarial Network (GAN) [8]. Recent SOTA combines both techniques, i.e. creating domain-invariant features which are based on source images translated to target images.

Real-to-sim domain adaptation [32] adapts the real images to synthetic images to make the robot feel at home for its navigation task. They use a cycleGAN [33] and shift loss for more consistent subsequent frames. The previous method translates every synthetic frame to the realistic style during the training of navigation policies. Although effective, this approach still adds an adaptation step before each training iteration, which can slow down the whole learning pipeline. Instead of using a GAN to upsample the image to perform domain adaptation, it is also possible to change the extracted features to another domain without upsampling. In French et al. [6] self-ensembling is used with a student-teacher method to achieve SOTA results on different visual domain adaptation benchmarks for classification. Adversarial Discriminative Domain Adaptation (ADDA) [29] generalizes the model from the source to the target domain by changing the feature encodings in the layer before the output layer. The main method used throughout this paper is based on ADDA and will be further explained in Section 3. More recently, the strong-weak alignment method [25] adapts global and local features adversarially with a domain classifier to again create domain-invariance. Selective Cross-Domain Alignment [34] uses a similar idea but focuses on discriminative regions of the image representations to perform adaptations. The main ideas are ”where to look” and ”how to align”. Diversify and match [12] obtains better generalization to other domains by diversifying the labeled data and then matching the features adversarially to make them close to domain-invariant. FRCNN in the wild [3] uses the representation from the RPN to get instance-level invariance and the image representation to get image-level invariance. Hsu et al. [9] combine techniques from ADDA and image-adaptation, using a cycleGAN, to improve generalization to a target domain.

All the methods listed above use a two-stage detector, mostly Faster-RCNN. These two-stage detectors reduce halfway the changes in performance introduced by the domain gap between the model trained on source data and target data (oracle). To the best of our knowledge, these unsupervised domain adaptation approaches have not been tested on one-stage detectors like YOLO or SSD which would significantly improve the fps and could enable use on edge devices.

3 Proposed Method

We hypothesize that a model trained on a large labelled dataset can be transferred to a new environment by adversarial training on- or offline. This hypothesis is based on the success of the transfer learning techniques mentioned in Section 2.2. We test this hypothesis by using the principle of adversarial feature manipulation for domain adaptation. Adversarial Discriminative Domain Adaptation [29] is a method to acquire a classification model for a target domain which only has unlabelled samples. This method consists of the following steps (see Figure 1): First, an initial classification model is trained on a large dataset of labelled data sampled from the source domain. Then, a domain discriminator and another classification and detection model, i.e. the target model, are trained alternately. The input of the discriminator is the feature encoding just before the last YOLO-layer, computed by alternately encoding the source and target images.

After training, the discriminator should not be able to distinguish the extracted feature encodings from source and target domain. This can be done by using an inverted-label GAN loss, with the following loss function for the domain discriminator:

$$L_{disc} = -(1 - Y)log(1 - D(E(I))) - Ylog(D(E(I)), \quad (1)$$

where Y represent the domain label, $E(I)$ the encoded feature from image I, and $D(X)$ the prediction of the domain classifier with feature X as input. The discriminator is trained by minimizing L_{disc}, while the encoder is trained by minimizing the binary cross-entropy loss of the detector and maximizing L_{disc}.

Finally, the target encoder is evaluated by feeding target samples which are mapped to an approximately domain-invariant feature space and afterwards classified by the source classifier

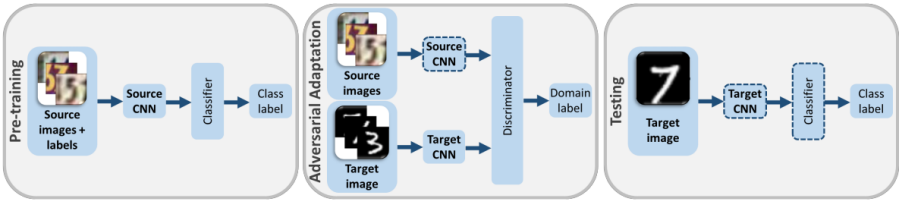


Fig. 1: Adversarial Discriminative Domain Adaptation (figure adapted from [29]) consists of three steps: 1. Pretraining the source model. 2. While freezing the source encoder, adversarially training target encoder and domain discriminator to obtain feature encodings that fool the domain discriminator. 3. Evaluate performance by combining target encoder and source classifier.

3.1 Adversarial Domain Adaptation for Object Detection

The focus in this paper is not only a classification task but also a localization aspect, i.e. object detection. Although different, the principle of ADDA in an object detection setting remains the same by mapping the feature encodings to a shared feature space between domains. As mentioned in section 2, the application of ADDA for object detection has been studied in several manners in conjunction with a Faster-RCNN network and has shown promising results. For successful domain adaptation with a one-stage detector like YOLOv3 [22], an extra step is needed to align the domain-invariant features with the source output layer, i.e. a detection and classification YOLO-layer. This can be done by training on a small target dataset for a couple of epochs. Consequently, there is no mismatch between encoding and output layer. The feature encodings will be slightly shifted to the target domain which can cause a decrease in performance in the source domain, yet enhance its performance on the target domain.

Applying ADDA in the context of the object detection task consists, mainly, of three steps (see Figure 2):

First, the source model needs to be fine-tuned on the large source dataset.

Second, (a) as an intermediate improvement step, it is possible to use a cycle-GAN [33] to create synthetic images from the source dataset that are more similar to the target images. This is achieved by using cycle consistency loss, which enables the use of unpaired data. Two generators map domain A to B and vice versa. The principle here is that by applying both generators sequentially, the output image should be the same as the input image. The comparison between input and output is the basis for the generator loss function. In between generators, domain classifiers differentiate between synthetic and real images. The generators are thus trained by minimizing generator loss and maximizing discriminator loss. In this way, we create an intermediate domain that is closer to the target domain, and makes it easier to close the domain gap with domain adaptation. These synthetic images substitute the original source dataset and do not alter the structure of the adversarial feature adaptation algorithm.

(b) Training a domain classifier alternately on source and target images to distinguish between them and adapt the feature maps with an inverted-label GAN loss [29]. This loss is used to achieve domain-invariant features in order to fool the discriminator. Important for this step is that the discriminator is pre-trained, otherwise, it may take a longer training time to show significant improvement, if any. The quality of the domain-invariant features depends on the quality of the domain classifier.

Third, the target model is fine-tuned with a small number of target images. For our experiments, we use 30 randomly chosen target images as the small fine-tuning dataset. More details will be presented in Section 5.2.

4 Implementation details

In our experiments we considering the YOLOv3 [22] detector with Darknet-53 feature extractor which has 53 convolutional layers. The input images are resized to 640×320 pixels and training is done with a batch size of 16. These features form the basis for the detection, classification and localization with the final YOLO-layer. Due to a feature pyramid network (FPN) [17], it is possible to predict objects more accurately at different scales because of the up- and downsampling steps with skip connections between layers with equal feature size. These skip connections combine low-resolution complex features with simple high-resolution features. The FPN of YOLOv3 consists of three different scaling stages and can thus predict for 3 different image sizes.

The domain classifier is a feed-forward model with 5 convolutional layers and a sigmoid classification layer at the end.

All models are trained on an Nvidia Tesla V100-SXM3-32GB GPU. For the training of the Darknet-53 network, binary cross-entropy is used as the loss function. Stochastic gradient descent with Nesterov momentum $\beta = 0.937$ optimizes training. The initial learning rate is $\alpha = 1\text{e}{-2}$ and the final learning rate is $\alpha_f = 5\text{e}{-4}$ where the learning is rate is defined by a cosine curve.

$$\alpha_{current} = \alpha_f + \frac{1}{2}(\alpha - \alpha_f)(1 + cos(\frac{epoch_{current}}{epoch_{max}}\pi) \tag{2}$$

The discriminator is trained on batches of 16 images utilizing an Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to decrease the binary cross-entropy loss function. The

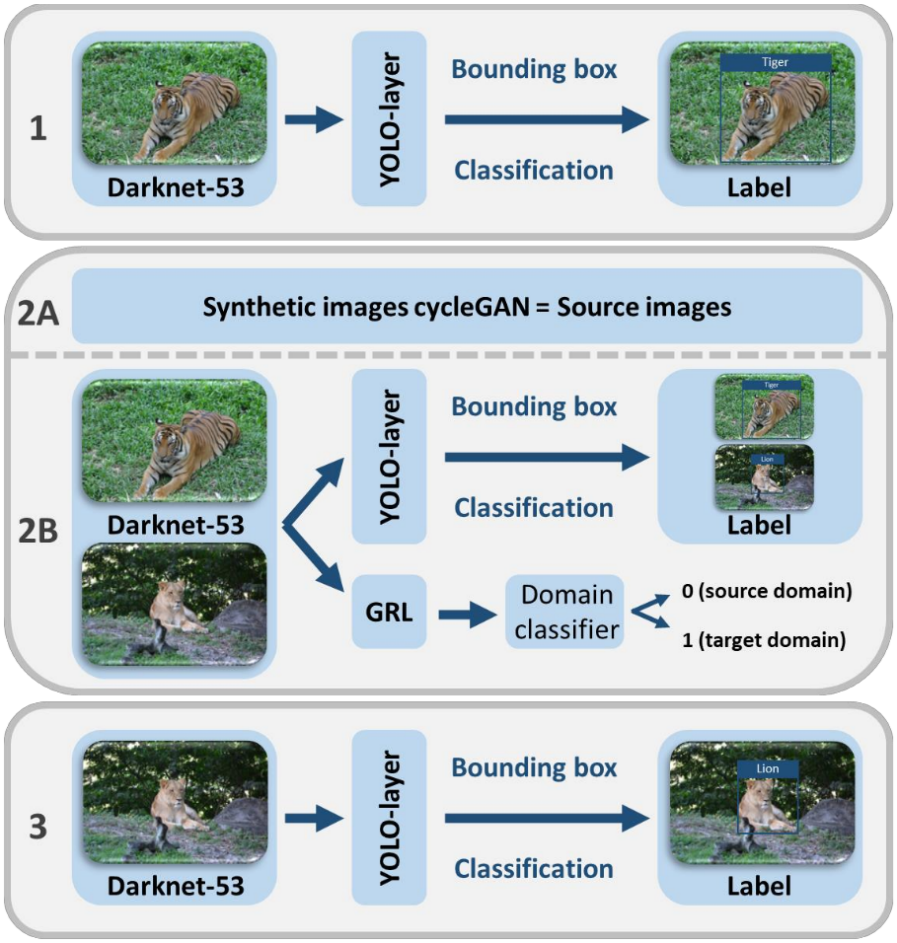


Fig. 2: Inspired by ADDA [29], our domain adaptation algorithm for one-stage object detection also consists of three steps: 1. Pre-training on source images. 2a. An intermediate improvement step of the model is replacing the source domain images with synthetic images generated from a cycleGAN. These generated images create an intermediate domain, which is closer to the target domain. 2b. Adversarially train Darknet-53 encoder and domain discriminator for obtaining domain-invariant features. Note that, the difference with ADDA here is that the source and target encoder have shared weights which improves generalization to both domains. 3. Fine-tuning the model on a small target dataset.

domain classifier is more difficult to train. Its learning rate is changed depending on the problem and domain gap. We empirically determined the interval of the learning rate as $\alpha = 2\text{e}{-8} \sim 2\text{e}{-10}$. The learning rate for the experiments is optimized with a hyperparameter sweep.

5 Evaluation

5.1 Datasets

We evaluate our method in the following datasets:

COCO2017 (COCO) [18] is a large dataset that consists of 80 labeled classes. In this study, only the car class is considered. From these examples, 8000 images are used for training and 4000 for testing.

KITTI Object Detection Benchmark (KITTI) [7] is a large annotated dataset with 15000 images captured from a car-mounted camera. We used 5400 images for training and 1300 for testing.

OpenImages (OI) [14] is a dataset of 9M images annotated with image-level labels, object bounding boxes, object segmentation masks, visual relationships, and localized narratives. For the transfer learning task covered in this research, we chose two similar classes, i.e. Tiger and Lion. Each class has approximately 1000 samples after cleaning up the data.

Cityscapes (CS) [4] is a dataset that consists of 6 labeled classes from urban street scenes. 2976 images are used for training and 500 for testing. For domain adaptation benchmarks, Cityscapes also has foggy Cityscapes dataset which consists of the same images synthetically augmented with a fog using depth images to blur distant objects [26].

5.2 Experiments

To validate the proposed adversarial feature adaptation method when integrated with single-stage detectors, we test our approach on two simple domain adaptation problems with datasets that look similar, i.e. a smaller domain gap. Concretely, we look at the COCO dataset versus the KITTI dataset with the focus on the car class. This exhaustively studied case can be a stepping stone towards other autonomous means of transportation. We will adopt the Mean Average Precision (mAP), at 0.5 Intersection over Union (IoU), precision, and recall as performance metrics. In addition, we report the framerate, i.e. the number of frames processed per second (fps), as an indicator of the computation costs during inference.

Inference speed The framerate is only dependent on the type of detector that is used as a backbone. The YOLOv3 detector achieves a framerate of 156 fps on an Nvidia Tesla V100-SXM3-32GB GPU and 1.83 fps on a 2.7GHz vCPU. In comparison, the Faster-RCNN with VGG-16 [28] achieves a framerate of 17 and 0.24 fps on the same GPU and vCPU, respectively. The latter is more representative for edge devices.

These results stress the need for good object detection performance with one-stage detectors since this speed-up can determine the feasibility of an application or not. For example, in the marine sector, a lot of research is done on autonomous vessels, with an operating speed from 5 to 15km/h. To navigate autonomously, they need to detect nearby objects in the waterway. Importantly, these vessels should be able to scan the environment frequently, around one or two times per second, which is sufficient at these low speeds [2]. Comparable new applications will thus benefit from a fast domain adaptation pipeline.

Object detection performance with domain adaptation To measure the performance of the method, we compare the mAP, precision, and recall of each transfer learning technique to a target domain with two models trained on target data. We compare with the following two models: *Base* (no TL): the vanilla YOLOv3 model trained on 30 annotated target images without transfer learning. *Oracle*: the YOLOv3 detector trained on the full annotated target dataset.

Table 1: Performance baselines on COCO and KITTI focused on the car class

fine-tuned on \ tested on	KITTI			COCO		
	mAP	P	R	mAP	P	R
COCO	0.744	0.623	0.785	0.704	0.666	0.713
KITTI (Base)	0.728	0.733	0.718	0.318	0.464	0.367
KITTI (Oracle)	0.974	0.936	0.961	0.22	0.822	0.166

Table 1 shows the results for the *Base* and *Oracle* models evaluated on the validation sets of the COCO and KITTI datasets. Several observations can be made in Table 1. First, and as expected, the models trained on images with the same distribution as the validation images have the highest performance. Second, a model fine-tuned on a larger dataset (*Oracle*) performs better than one trained on a smaller dataset (*Base*). However, it seems that this gain in performance comes a the cost of lower generalization towards other datasets, e.g. cross-dataset evaluation.

Taking the observations from above into account, we expect the performance of the models with transfer learning to lie somewhere in between *Base* and *Oracle*.

Measuring the effect of domain adaption We designed two experiments to measure the effect that domain adaptation has on performance.

On the first experiment we apply domain adaptation to train a model from COCO to KITTI dataset, with the focus on the car class. This experiments will focus on modelling intra-class variations introduced by the domain shift.

On the second experiment, we apply domain adaptation to train a model for a Lion class starting from the Tiger class. Both classes are extracted from the OpenImages dataset. Here, transfer learning is performed between two similar, yet different, classes.

This experiment aims at assessing the effect of domain shift caused by inter-class variations [10].

For both experiments, we limit ourselves to only consider 30 labelled target images. In future work, the minimum number of needed labelled target images to achieve improvements will be investigated.

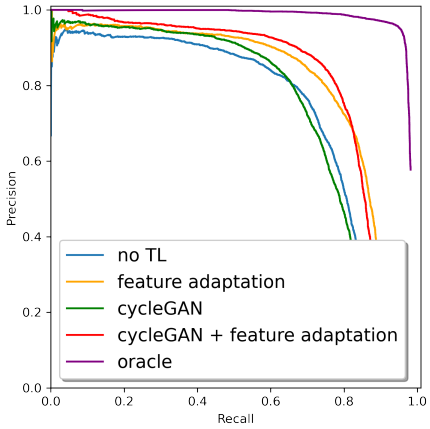


Fig. 3: Precision and Recall curves for the different transfer learning methods (trained on the COCO dataset) and the oracle model, evaluated on the KITTI dataset.

Table 2: Performance domain adaptation techniques from COCO to KITTI tested on both COCO and KITTI validation sets.

	KITTI			COCO		
Method	mAP	P	R	mAP	P	R
No TL (*Base*)	0.728	0.733	0.718	0.318	0.464	0.367
Feature adaptation	0.796	0.824	**0.727**	0.584	0.729	0.54
CycleGAN	0.733	0.81	0.653	0.421	0.664	0.378
CycleGAN + feature adaptation	**0.797**	**0.876**	0.714	0.519	0.826	0.411

Tables 2 and 3 show the results for experiments one and two, respectively.

In general, it can be noted that combining feature adaptation with synthetic data augmentation from a cycleGAN gives the best results (mAP) for both experiments in their respective target domains, i.e. KITTI-cars and Lion. The models for KITTI-cars and Lion improve 5% and 10%, respectively, compared to their *Base* performance. If we define the domain performance gap as the difference in mAP between *Base* and *Oracle*, then the gap is closed by 28%, from 0.728 to 0.797 with an *Oracle* of 0.974 mAP.

Fig. 4: The ground truth is shown above the line and predictions on images from the KITTI dataset, generated by the different models, under the line. The models from top to bottom are: no TL (Base) (R1), feature adaptation (R2), cycleGAN (R3), feature adaptation with cycleGAN (R4), and Oracle (R5).

Table 3: Performance domain adaptation techniques from Tiger to Lion tested on both Tiger and Lion validation sets.

	Lion			Tiger		
Method	mAP	P	R	mAP	P	R
no TL	0.727	0.919	0.609	0.797	0.915	0.607
Feature adaptation	0.764	0.855	**0.715**	0.908	0.881	0.906
CycleGAN	0.747	**0.99**	0.661	0.947	0.967	0.836
CycleGAN + feature adaptation	**0.768**	0.922	0.711	0.926	0.896	0.906

The precision improves significantly by adding the synthetic images while maintaining a similar recall. Figure 3 further confirms the fact that the combination of feature adaptation together with synthetic images from a cycleGAN has the best performance out of the domain adaptation techniques. It also shows that using synthetic images has an advantage regarding precision while maintaining a similar recall. As hypothesized earlier, we observe that the *Oracle* model outperforms the transfer learning techniques.

The focus is on the car experiment, as the KITTI and COCO dataset sizes are large compared to the OI datasets of the Tiger and Lion classes. This means that more labelled source domain images are present for training, and the evaluation results are more accurate representations of the models' performance on the target domain, as an outlier will have less impact on the overall performance. Although smaller, the Tiger to Lion domain adaptation still shows the increased performance with adversarial learning in an inter-class setting.

Figure 4 shows qualitative detection results. More specifically, it shows predictions of the different models on the KITTI validation dataset. The different baselines include: *Base* (no TL) (R1), a feature adaptation model (R2), a model trained on cycleGAN synthetic images (R3), a feature adaptation model with cycleGAN synthetic images (R4), and *Oracle* (R5). The target models are designed for the target dataset. Remarkably, applying transfer learning techniques improves the generalization back to the source domain. This is in contrast to no transfer learning (no TL) with a model only fine-tuned on 30 labelled target images starting from the source model. The domain-invariant features and the intermediate domain dataset generated from a cycleGAN play the most important factors for this result.

Use of intermediate domain from a cycleGAN Figure 5 shows the result of using a cycleGAN to generate the synthetic images in an intermediate domain between source and target domain. It is clear that after the transformation from tiger to lion, the tiger stripes have vanished and that the colour changed from orange to tawny yellow. There is a blurring effect that can have a negative effect on performance but this is likely caused by the small size of the Lion dataset. The transformation from COCO to KITTI mostly changed the background as the COCO dataset contains more urban-based images while the KITTI dataset depicts cars more in or around a forest. That is why the generated images contain fake trees in the background, even in the reflection of the car window.

Effect on the domain-shift The accuracy of the domain classifier, before and after adversarial training on the image encoder, can also provide some insight on the observed performance. Before any adversarial training of the feature encodings, the pre-trained domain classifier can predict with approximately 55% accuracy, in both experiments, what the domain of the tested feature encodings is. After adversarial training, this drops to 50%. The similarity between datasets causes a very low accuracy of 55%. Still, the feature encoder manages to extract useful information from the domain classifier to compensate for the subtle differences between datasets.

To follow up on this observation, we conducted an additional experiment focused on the ships class. More specifically, where the source dataset is the Seaships dataset [27] and the target dataset is a self-annotated dataset from videos recorded on a cargo ship on inland waterways. The main difference between those two datasets is the point-of-view, on-board versus on-shore. Because of this significant difference, the discriminator model performs very well and has an accuracy of 95+%. Because of this large domain gap, the adversarial model is not able to manipulate the encoded feature spaces toward each other. This shows the limitations of using *only* adversarial training. More pre-processing steps are needed than only a Cycle-GAN to close the domain gap for effective adversarial learning.

Unsupervised setting In Table 4 a comparison is made between existing methods and the methods explained and tested in this paper in an unsupervised manner to adapt from the Cityscapes to the foggy Cityscapes dataset. The difference with the experiments above is that this time, there is not a last fine-tuning step with a small target dataset. In Table 4, it is clear to see that the methods with feature adaptation in combination with YOLO do not improve the results. Using a CycleGAN to create synthetic images works well. As the foggy Cityscapes itself is a synthetic dataset, it is not surprising that training on synthetic images from a CycleGAN generates a good result.
The other methods all also use some kind of adversarial feature adaptation, the main difference is the object detection architecture. In Faster-RCNN, there is a Region Proposal Network (RPN) which already gives a good idea where objects of interest are while filtering out the background. Our theory is that performing adversarial feature adaptation on these region proposals is much more specific and accurate domain adaptation. This understanding can be the key for future work to understand how to replace this RPN in YOLO to have fast, accurate and specific domain adaptation without the need for a small target dataset. The previous semi-supervised experiments are still valid as they improve the baseline model significantly.

Summarizing To summarize, this one-stage object detection model enables near real-time use on edge devices with 2 fps on a 2.7 GHz vCPU. The domain performance gap is reduced by 28% (difference between mAP of *Base* and *Oracle*) on the COCO (source) and KITTI (target) datasets. The synthetic images from a cycleGAN to replace the source images have a positive effect on the precision and mAP of the model and form a good option to boost performance. The algorithm works both for inter- and intra-class domain adaptation.

Table 4: Performance domain adaptation techniques from Cityscapes to foggy Cityscapes, tested on the foggy Cityscapes validation set.

Method	car	truck	bus	train	motorcycle	bicycle	mAP
FRCNN in the wild [3]	40.5	22.1	35.3	20.2	20.0	27.1	27.6
Diversify and Match [12]	44.3	27.2	38.4	34.5	28.4	32.2	34.6
Strong-Weak Align [25]	43.5	24.5	36.2	32.6	30.0	35.3	34.3
Progressive DA [9]	54.4	24.3	44.1	25.8	29.1	35.9	36.9
Feature adaptation	45.9	26.9	22.1	4.77	12.3	21.8	22.3
CycleGAN	68.7	41.8	40.1	17.9	16.7	30	35.9
CycleGAN + feature adaptation	37	27.5	30.4	14.2	7.46	16	22.1

6 Conclusion

We presented a method that enables object detection with a limited amount of labels on edge devices in near real-time. The main advantages are three-fold. First, the use of only a limited annotated target dataset, the amount of labels needed depends on the desired trade-off between cost and performance. Second, by using a one-stage detector, the proposed systems achieves an increased object detection speed approximately eight times faster. This enables the possibility to use edge devices, such as a 2.7GHz CPU which reaches almost 2fps. Third, a reduction of 30% in the changes in performance introduced by the domain gap. Moreover, we observed a significant increase in performance for inter- and intra class domain adaptation. In the unsupervised setting, we saw that finding an alternative for the RPN, implemented in the Faster-RCNN model, for the YOLO model can accelerate the adversarial training to achieve specific, accurate and fast domain adaptation. There are also some disadvantages of using this method: On the one hand, a two-stage detector like Faster-RCNN closes the domain gap more. In Hsu et al. [9] the domain gap is closed by 56% where the target domain is Cityscapes [4] and the source domain is KITTI, also focused on the car class. On the other hand, a source domain with abundantly available data is needed that resembles the target domain. In our experiment, these source domains are the Tiger class and COCO. When the gap is too large between source (Seaships) and target domain (self-annotated vessel dataset), using only adversarial training methods fall short and additional preprocessing is needed to close the domain gap before using this algorithm.

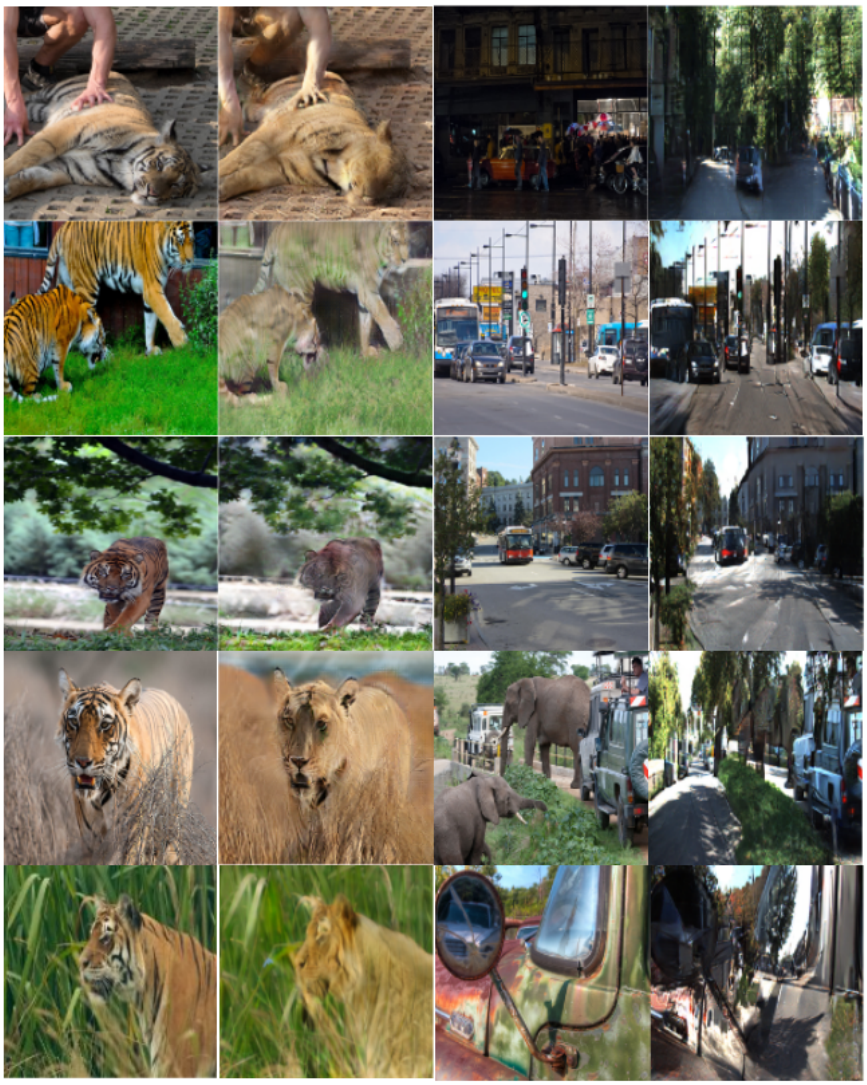


Fig. 5: These four columns of images show the transformations, by using a cycleGAN, of the source domain images to generate synthetic images, which try to match the target domain distribution. The source image is shown in the first and third column in both examples (Tiger from Open Images, and car from COCO), and the generated output which tries to mimic the target images is shown in the second and fourth column (fake Lion from Open Images, and fake car from KITTI). In the Tiger to Lion example, the generated output is blurred, yet tiger stripes have vanished and the colour changed from orange to tawny yellow. In the car example, the environment changes from urban to woodland.

Bibliography

[1] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Computer Vision and Image Understanding **110**(3), 346–359 (2008). https://doi.org/https://doi.org/10.1016/j.cviu.2007.09.014, https://www.sciencedirect.com/science/article/pii/S1077314207001555, similarity Matching in Computer Vision and Multimedia

[2] Bole, A., Wall, A., Norris, A.: Chapter 1 - basic radar principles. In: Bole, A., Wall, A., Norris, A. (eds.) Radar and ARPA Manual (Third Edition), pp. 1–28. Butterworth-Heinemann, Oxford, third edition edn. (2014). https://doi.org/https://doi.org/10.1016/B978-0-08-097752-2.00001-5, https://www.sciencedirect.com/science/article/pii/B9780080977522000015

[3] Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)

[4] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

[5] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 886–893 vol. 1 (2005). https://doi.org/10.1109/CVPR.2005.177

[6] French, G., Mackiewicz, M., Fisher, M.H.: Self-ensembling for domain adaptation. CoRR **abs/1706.05208** (2017), http://arxiv.org/abs/1706.05208

[7] Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)

[8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)

[9] Hsu, H.K., Hung, W.C., Tseng, H.Y., Yao, C.H., Tsai, Y.H., Singh, M., Yang, M.H.: Progressive domain adaptation for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2019)

[10] Kang, G., Jiang, L., Yang, Y., Hauptmann, A.G.: Contrastive adaptation network for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

[11] Khirodkar, R., Yoo, D., Kitani, K.: Domain randomization for scene-specific car detection and pose estimation. In: 2019 IEEE Winter Confer-

ence on Applications of Computer Vision (WACV). pp. 1932–1940 (2019). https://doi.org/10.1109/WACV.2019.00210

[12] Kim, T., Jeong, M., Kim, S., Choi, S., Kim, C.: Diversify and match: A domain adaptive representation learning paradigm for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

[13] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)

[14] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., Ferrari, V.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. IJCV (2020)

[15] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4), 541–551 (1989). https://doi.org/10.1162/neco.1989.1.4.541, https://doi.org/10.1162/neco.1989.1.4.541

[16] Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Proceedings. International Conference on Image Processing. vol. 1, pp. I–I (2002). https://doi.org/10.1109/ICIP.2002.1038171

[17] Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)

[18] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014)

[19] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)

[20] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)

[21] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)

[22] Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. CoRR **abs/1804.02767** (2018), http://arxiv.org/abs/1804.02767

[23] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)

[24] Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International conference on computer vision. pp. 2564–2571. Ieee (2011)

[25] Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

[26] Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. International Journal of Computer Vision **126**(9), 973–992 (2018)
[27] Shao, Z., Wu, W., Wang, Z., Du, W., Li, C.: Seaships: A large-scale precisely annotated dataset for ship detection. IEEE Transactions on Multimedia **20**(10), 2593–2604 (2018)
[28] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
[29] Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. CoRR **abs/1702.05464** (2017), http://arxiv.org/abs/1702.05464
[30] Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E.: Deep learning for computer vision: A brief review. Computational intelligence and neuroscience **2018** (2018)
[31] Wu, B., Iandola, F., Jin, P.H., Keutzer, K.: Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (July 2017)
[32] Zhang, J., Tai, L., Yun, P., Xiong, Y., Liu, M., Boedecker, J., Burgard, W.: Vr-goggles for robots: Real-to-sim domain adaptation for visual control. IEEE Robotics and Automation Letters **4**(2), 1148–1155 (April 2019). https://doi.org/10.1109/LRA.2019.2894216
[33] Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
[34] Zhu, X., Pang, J., Yang, C., Shi, J., Lin, D.: Adapting object detectors via selective cross-domain alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

Task Independent Capsule-based Agents for Deep Q-Learning

Akash Singh, Tom De Schepper, Kevin Mets, Peter Hellinckx, José Oramas, and Steven Latré

University of Antwerpen, imec IDLab, Antwerpen, Belgium
akash.singh@uantwerpen.be
https://www.uantwerpen.be/en/

Abstract. In recent years, Capsule Networks (CapsNets) have achieved promising results in tasks such as object recognition thanks to their invariance characteristics towards pose and lighting. They have been proposed as an alternative to relational insensitive and translation invariant Convolutional Neural Networks (CNN). It has been empirically proven that CapsNets are capable of achieving competitive performance while requiring significantly fewer parameters. This is a desirable characteristic for Deep reinforcement learning which is known to be sample-inefficient during training. In this paper, we propose DCapsQN, a task-independent CapsNets-based architecture in the deep reinforcement learning setting. We experiment in the model-free reinforcement learning setting, more specifically in Deep Q-Learning using the Atari suite as the testbed of our analysis. To the best of our knowledge, this work constitutes the first CapsNets-based deep reinforcement learning architecture to learn state-action value functions without the need for task-specific adaptation. Our results show that, in this setting, DCapsQN requires 92% fewer parameters than the baseline. Moreover, despite their smaller size, the DCapsQN provides significant boosts in performance (score), ranging between 10% - 77% while further stabilising the Deep Q-Learning. This is supported by our empirical results which shows that DCapsQN agents outperform the benchmark Double-DQN agent, with Prioritized experience replay, in eight out of the nine selected environments.

Keywords: Deep reinforcement learning · Capsule networks · Deep Q-learning.

1 Introduction

Reinforcement Learning (RL) is an experience-based learning paradigm, where the agent interacts with the environment by performing an action and learns how to maximize its cumulative reward based on the returned rewards. The learning is based on trial and error and often requires a large amount of data for Deep Reinforcement Learning (DRL). In recent years, with advancements in Deep Learning (DL), Convolutional Neural Networks (CNNs) have made breakthroughs in multiple machine learning tasks like natural language processing

and computer vision [12, 14]. The field of DRL has benefited from the remarkable flexibility and advancement of DL as well. CNNs have remarkable flexibility to learn features for the agent to learn a proper policy or value function. Having scalar nature, CNNs have additive nature in neurons at any given layer, they are ambivalent to spatial relationships within their kernel of previous layers [15]. Thus despite their good performance, they have an inherent weakness of limited modelling capabilities for spatial relationships between the learned features [25, 29]. For example, for the task of recognizing faces in images, CNNs are capable of learning the regions that resemble a nose or a mouth. However, when recognizing a face, at test time, they have the weakness of focusing on the occurrence of these "facial parts" and completely ignore the spatial arrangement in which these should occur in order to effectively represent a face.

Capsule Networks (CapsNets) were designed to mimic human vision [9, 25]. They address the inherent limitation of CNNs, while significantly decreasing the required number of parameters. CapsNets aim to preserve the spatial information (pose and precise location) and attributes (length, thickness etc) by encoding features in vectors rather than scalar values. Under this formulation, the magnitude of the vector represents the probability of the existence of the entity it is representing. CapsNets in DL require less training data, which is a desirable attribute within a DRL setting. The architectural design of CapsNets profits from *dynamic routing*. Routing by agreement is a novel dynamic routing technique, it plays a key role in preserving spatial information. The architectural overview of capsules draws inspiration from the Multi-Layer Perceptron architecture. This architecture with *routing by agreement* is designed to preserve part-whole relationships (locations, orientations, etc.) between various entities levels which may be a complete entity or part-of an entity. For example, the relative positions of a nose and a mouth on a face in a portrait. [25] used the magnitude of a vector from the last layer of CapsNets for classification in supervised deep learning.

Reinforcement learning approaches such as DQN strive to estimate the action-value function [19, 18]. Traditionally for vision-based tasks, an agent's architecture uses CNNs and fully connected layers to approximate the optimal action-value function. The CNN-based architecture of the agent in various deep reinforcement learning algorithms [19, 28, 26] are inspired from [11]. The agent learns on raw sensory input that uses CNNs to mimic the effects of receptive fields [19]. While the magnitude of the vector in CapsNets is a good surrogate for multi-class classification, it is not a good candidate for estimating the state-action value function in DRL.

Here we propose DCapsQN, an architecture suitable for an agent to learn value functions based on part-whole relationships. We demonstrate how part-whole relationships assist in value function estimation and that Q-estimates from them are much more self-coherent. Owing to a large number of atari environments and their experimental/computational costs, we limit our experiments to a diverse subset of environments with different natures and tasks.

Across multiple environments, the proposed agent uses 92% fewer parameters and improves 10%-77% on performance (score) compared to the baseline.

The main contributions of this paper are:

1. Introducing DCapsQN, a task-agnostic CapsNets-based architecture.
2. Presenting the first CapsNets based architecture study on the atari benchmark.
3. Comparing DCapsQN to the traditional CNN-based architecture of DeepQN, showing a reduction in the number of trainable parameters.

2 Background

2.1 Capsule Networks

Computer graphics employ *Hierarchical Modeling* for building complex objects by placing simpler objects and their known relations [7]. The idea of CapsNets is to achieve the capabilities of inverse hierarchical modelling to better understand the scene where lower level capsules represents simpler entities and higher level capsule represent the complex. The concepts of capsule (Fig. 1) and CapsNets (Fig. 2) were introduced in [25] to retain the spatial relationship between complex and simple entities [9, 25]

CapsNets architecture is inspired from Multi-Layer Perceptron architecture, where a capsule replaces a neuron in a layer. Capsule [25], as a fundamental unit of CapsNets, can be defined as a group of neurons where the activities of the neurons within a capsule represent the various properties like pose (position, size, orientation) (Fig. 1). Capsule encodes an entity as a vector where its magnitude represents the probability of entity occurrence and its orientation represents attributes of the entity (Fig.1). The magnitude of the vector output is always bound between 0 and 1.

We arrange capsules in 2 levels, in lower level l they are called primary capsules and upper-level $l+1$ they are called secondary capsules.

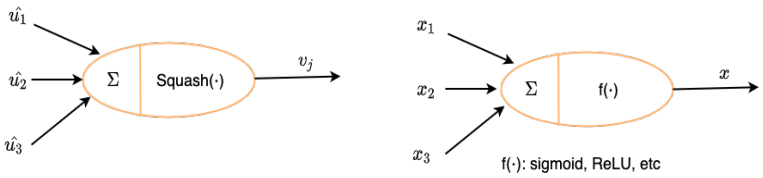


Fig. 1: The similarity between a capsule and a neuron. [16].

Primary capsules: Following the first convolutional layer, the primary capsule (*PrimaryCaps*) is responsible for transforming scalar values into a vector. A capsule in Fig.2 refers to a group of convolutional layers. It is the first layer where the process of inverse hierarchical modelling takes place. The capsule here reshapes the feature maps outputs of convolutional layers to output vectors.

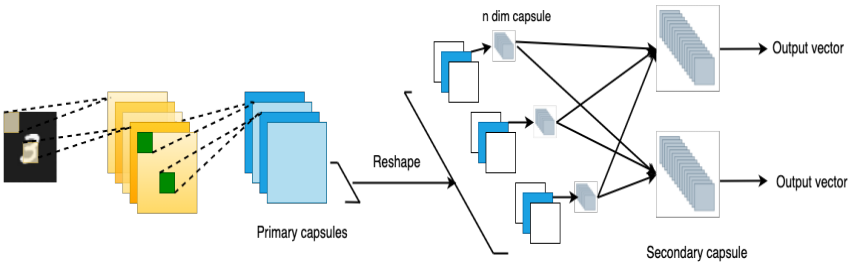


Fig. 2: The figure shows fundamental Capsule network architecture.

Secondary Capsules: Following PrimaryCaps is Secondary Capsules (*SecondaryCaps*). They receive an input vector from PrimaryCaps. The weight matrix $\mathbf{W}_{ij}$ transforms the output vector of a PrimaryCaps to serve as input to SecondaryCaps.

$$\hat{u}_{j|i} = W_{ij} u_i \tag{1}$$

Routing by agreement: Routing by agreement is a dynamic routing technique introduced in [25]. Pooling operations statically forward the relevant information from the previous layer to the following layer and in this process, it loses information. Contrary to statically connected pooling layers, dynamic routing during the forward pass redirects the output from PrimaryCaps to the most relevant parent in SecondaryCaps. Each capsule i (where $1 \leq i \leq N$) in a layer l has vector u_i to encode spatial information. The output of PrimaryCaps u_i of the ith layer acts as input to all capsules in layer $l+1$ of SecondaryCaps.

The *Coupling coefficient* c_{ij} is iteratively determined through routing by agreement. It represents the agreement of a capsule of layer l with $l+1$. If the agreement is high, the coupling coefficient for child-parent will increase, otherwise, it would decrease. The coupling coefficient plays a role in the child-parent relationship to form a parse tree-like structure in CapsNets. The weighted sum (s_j) from all PrimaryCaps contributes to forming the output of SecondaryCaps.

$$s_j = \sum_{i=1}^{N} c_{ij} \hat{u}_{j|i} \tag{2}$$

The magnitude of the output vector from PrimaryCaps is limited between 0 and 1 by using a *squashing function*. The magnitude of the vector represents the probability of the existence of an entity represented by a capsule.

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2} \frac{\mathbf{s}_j}{\|\mathbf{s}_j\|} \tag{3}$$

The squashing function makes sure to limit the length while still retaining the positional information.

2.2 Deep Reinforcement Learning

We study the utility of CapsNets-based representations in Double DQN using prioritised experience replay. The method uses *proportional prioritization* of prioritised experience replay.

The Q-learning algorithm is a temporal difference learning algorithm. To update the value estimate of a state-action pair, the temporal difference (TD) error is computed at each time step. Deep Q-learning was first introduced by [18] to approximate Q-values for high dimensional sensory input. Deep Q-learning is known to be unstable and it overestimates the Q-values. To remedy this [28] proposed Double DQN. They decoupled the networks for selecting and evaluating an action separately. The agent generally selects an action using ϵ-greedy policy. Under the ϵ-greedy policy, agents can take a random action with ϵ probability or select an action with 1-ϵ probability maximising $Q(s, a)$.

An *Experience replay* is used to store the agent's interaction with the environment at each time step [18]. This buffer is used to sample batches of experience during training. [26] proposed a new experience replay design called prioritised experience replay (PER), where the most important experiences were replayed to the agent. The importance or priority of experience was calculated using the TD error. With the design choice, [26] were able to empirically show that experience replay became more efficient and effective, which led to even better and faster learning of an agent. The agent performed better compared to the previous state-of-the-art DQN.

3 Related work

On account of the drawbacks of CNNs, [25] introduced the idea of CapsNets, but most of the published research on CapsNets is currently focused in the field of deep learning.[5, 21, 23, 24] extend the work of [25] by proposing new capsule-based architectures. [23] proposes a DenseNet-like skip connection where the standard convolution component in the CapsNet is replaced with a hierarchical architecture. The resulting architecture outperforms the original CapsNets on datasets like SmallNORB and Cifar-10. [24] remove the margin loss to show that unsupervised training of sparse capsules can potentially lead to deeper architectures while achieving higher accuracy. [5] proposes a novel routing algorithm based on eigen-decomposition of votes. This leads to a higher convergence speed of the new architecture compared to original CapsNets. [2, 3, 1] and [15] investigate the performance of CapsNets in medical applications like brain tumour classification, COVID cases classification, Alzheimer disease detection and Lung segmentation. [30] study 3D-capsules for pose estimation. The work exploits the structural relations among local parts for pose estimation. [10] propose dual attention mechanism capsule network for higher accuracy and faster training.

While CapsNets have gained popularity in standard deep learning approaches, their study within a Deep Reinforcement Learning (DRL) context has received significantly less attention. [4] tries integrating CapsNets with Deep-Q Learning.

They showed that CapsNets-based agents underperform with respect to their baseline. The experiments were done on FlashRL with environments like Flappy Bird, Deep Line wars etc. The architecture takes 84×84 input which propagates to output $n \times 16$ vector from last capsule layer. n being number of actions. The architecture proposed by [4], employs the magnitude of the vector output from the last capsule layer for action-value estimation. The authors [4], do not take into consideration that magnitude of the vector from a capsule is not a good fit for action-value estimation. While the value function could have any negative or positive value, the magnitude of the vector output from CapsNets is bounded between zero and one (Eq.3).

[20] combines CapsNets with A2C, but limits the scope of the study to only maze navigation in the ViZDoom environment. The ViZDoom environment only [13] provides tasks like move-and-shoot and maze navigation. Unlike the ViZDoom, the atari benchmark offers a more diverse, challenging and conceivable tasks in learning, modelling, and planning. Inspired from previous studies [18, 28, 19], we choose a widely accepted Atari benchmark [6] to empirically show the advantage of our framework in task-agnosticism and parameter reduction. The study proposes a generalised CapsNets-based agent to learn a state-action value function with no task-specific adjustments. Our DCapsQN, to the best of our knowledge, is the first generalised, task agnostic framework to learn state-action value functions to solve nine diverse atari tasks in addition to maze traversals.

4 Methodology

In this section, we introduce the agents and the environment used as a testbed for the analysis. We employ the atari suite for our experiments as it provides a variety of environments with respect to input space, action space and rewards.

Baseline Agent For the baseline, we choose Double-DQN with prioritised experience replay [26, 28]. The first layer in this architecture is a convolutional layer composed of 32, 8×8 convolution kernels with a stride of 4. This first layer feeds a second convolutional layer of 64, 4×4 kernels with a stride of 2. The third layer receives input from the second and has 64, 3×3 kernels with a stride of 1. The last convolutional layer of this set is connected to two FC layers. The first FC layer is composed of 512 neurons while the second FC layer is composed of a number of neurons equal to the output value estimates for the actions of interest. ReLU acts as the activation function for all the layers except the last FC layer. The architectural design of the CapsNets-based agent is depicted in Fig. 3 (bottom).

DCapsQN Agent In a DRL agent, CNNs learn relevant visual features with respect to the task at hand while the FC layers aim at learning valuable combinations of these features and map them to value functions related to the actions of interest. In this regard, the FC layers learn the value function based on the features generated by CNNs. We explore the application and utility of CapsNets-based representations with Double DQN. The architectural design of

the DCapsQN depicted in Fig. 3 (top), takes inspiration from [25, 28] to learn part-whole relationship between visual entities in input state.

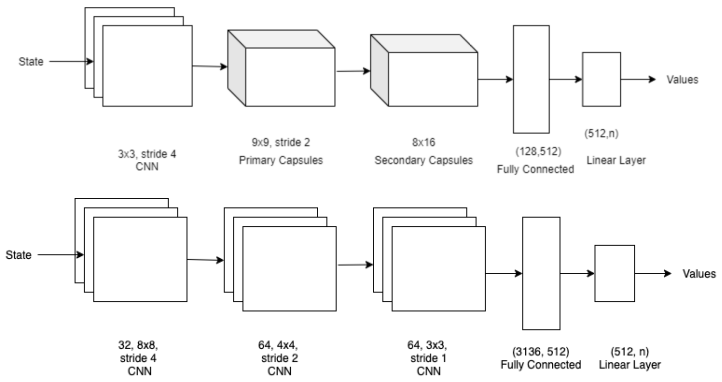


Fig. 3: DCapsQN (top) and Double-DQN (bottom) architecture.

A convolutional layer acts as the first layer, as shown in Fig.3. The Convolution layer has 16, 3×3 convolution kernels with a stride of 4 and ReLU activation. This layer detects features from states and serves as an input to the Primary capsule layer. We have 49 capsules in the Primary capsule layer. A Primary capsule layer, here is a collection of convolutional capsules. A single convolution capsule comprises of a group of convolution layers with 9×9 kernel and with a stride of 2. Each capsule in the PrimaryCaps receives the input of all convolutional layers. Each primary capsule outputs an 8-dimensional vector. The output from the Primary capsule serves as input to the Secondary capsule layer. The Secondary capsule layer has 8 capsules with each Secondary capsule producing a 16-dimensional vector as output. Each of the Secondary capsules receives the input from all Primary capsules. The connection between the PrimaryCaps layer and the SecondaryCaps is controlled by *dynamic routing*. In our study, we followed the *routing by agreement* algorithm [25] where each child chooses its parent based on the cosine similarity between its transformed vector output and the vector output of its candidate parent. The dynamic routing between layers utilizes the vector output from capsules to preserve hierarchical relations in a state. Three routing iterations are used between capsule layers in order to find optimal weights for relations between layers.

Environment The Arcade Learning Environment (ALE) [6] is a popular benchmark composed of a collection of Atari 2600 games. It provides a challenging and diverse set of tasks with respect to visual input, rewards returned by the environment, action space and difficulty. [17] integrate around 40 techniques from a dozen papers in order to determine the difficulty level of the games that are part of the benchmark.

Atari offers 57 environments, to compare the performance of our DCapsQN agent with respect to the baseline agent, we choose a subset of the environments that are diversified in terms of visual input (simple, complex), reward (sparse,

dense), action space(3, 4, 6, 8, 18) and difficulty score [17]. Across various tasks, both agents are tasked with collecting the maximum reward. The environment gets reset the moment when the agents use all of their lives.

The input states are composed of simple states such as Pong, Boxing to fairly complex input states like Fishing Derby or Alien. The tasks are also diversified with respect to rewards offered by the environments. The agents are evaluated with dense rewards environments like Breakout, Pong and sparse rewards environments like Fishing Derby. Further, we the select the tasks that lay in difficulty spectrum of -2 to 10. Higher the difficulty score, lower was the performance of most able techniques considered in [17]

Training protocol With Atari, we restrict the training of both agents to only 20 million steps. The DCapsQN-based agent uses a batch size of 128 and a Learning rate of 0.00015 with RMSprop optimizer and Prioritised experience replay with alpha = 0.5 and beta with linear annealing from 0.4 to 1. The other hyper-parameters such as discount rate, the size of the experience replay memory, target network updates are the same as [26]. Baseline agents use the same hyper-parameters as described in [26]. An epsilon-greedy action selection method is employed to balance our exploration and exploitation. Both Double DQN and DCapsQN based agents randomly explore for the first 50000 steps and then linearly decrease the probability to randomly select an action for the next 1e6 steps. At end of 20 million steps, there still remains an exploration probability of 0.01. The evaluation section compares the cumulative reward collected by agents in all tasks. The average is calculated from 4 randomly initialized agents.

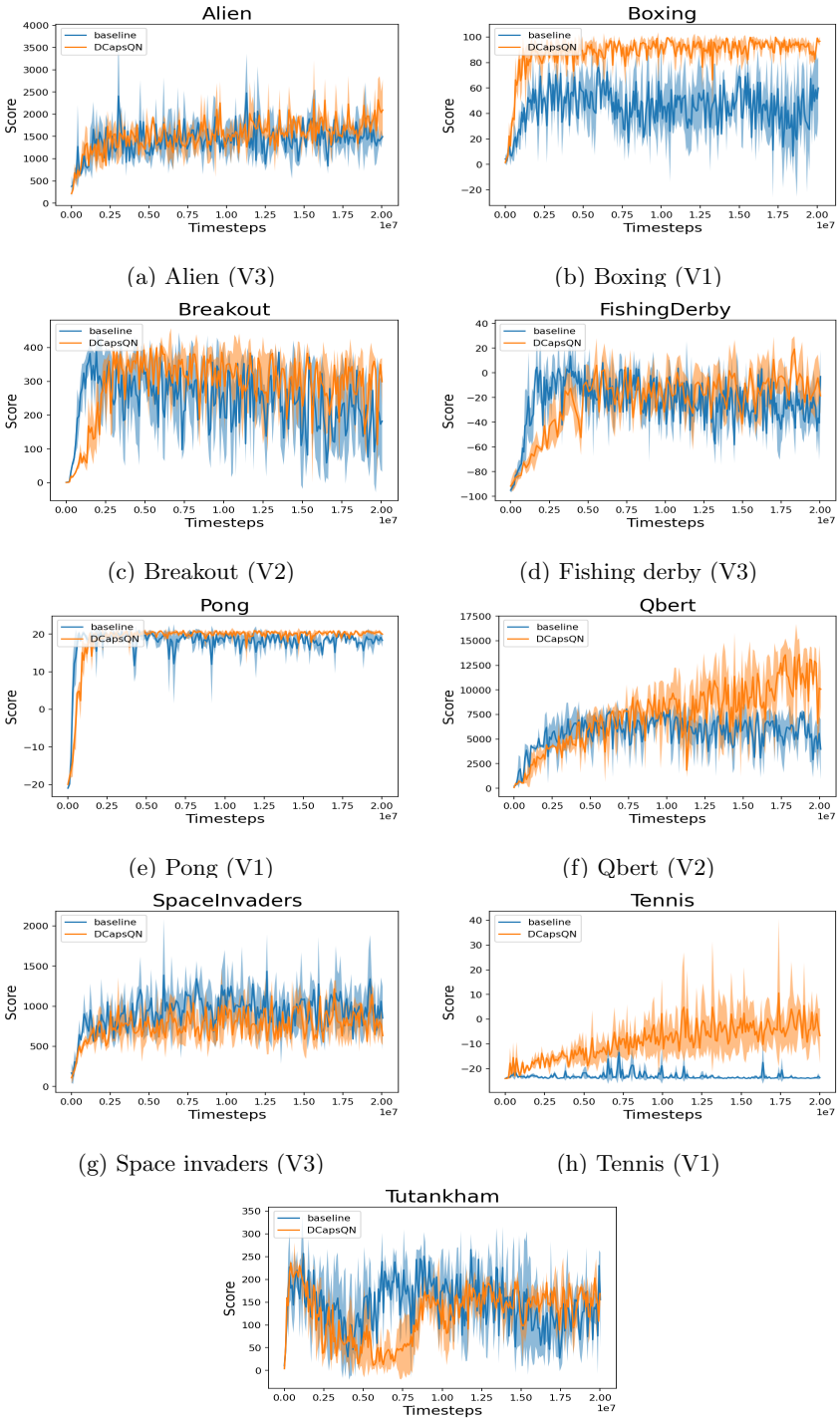
Evaluation protocol For evaluation, we refer to [26, 28]. We evaluate both agents every 1 million steps and average over 100 episodes. The other hyper-parameters are the same as Double-DQN [28].

5 Analysis

In any given task an agent collects rewards to maximize its performance. The cumulative reward collected by an agent is the attribute that links to the agent's success in a given task. Apart from the cumulative rewards, to better understand the CapsNets-based representation in DRL environments, we try to get a deeper insight regarding the agents' performance under different attributes, e.g. input states, rewards and action space, of the environments.

5.1 Cumulative reward and Parameters

Our DCapsQN agent (Sec 4) has around 92% lower number of trainable parameters compared to baseline. To highlight the difference, Table 1 presents a comparison of trainable parameters of both agents under different environments. To show the effectiveness of the representations learned via CapsNets, we compare the agents' performance with respect to the cumulative reward collected by them in all of the analyzed tasks. Table 2 presents the comparison of the performance of both agents. Though DCapsQN agents have a lower number



(a) Alien (V3)

(b) Boxing (V1)

(c) Breakout (V2)

(d) Fishing derby (V3)

(e) Pong (V1)

(f) Qbert (V2)

(g) Space invaders (V3)

(h) Tennis (V1)

(i) Tutankham (V3)

Fig. 4: Average score collected by the agents in the respective environment. The agents follow an epsilon greedy policy. The shaded area represents the ± standard deviation over 4 runs.

Table 1: Parameters comparison.

Environment name	DCapsQN parameters	Baseline parameters	Difference
Alien (V3)	**136,426**	1,693,362	91.94%
Boxing (V1)	**136,426**	1,693,362	91.94%
Breakout (V2)	**129,244**	1,686,180	92.33%
Fishing Derby (V3)	**136,426**	1,693,362	91.94%
Pong (V1)	**130,270**	1,687,206	92.27%
Qbert (V2)	**129,244**	1,686,180	92.33%
Space Invaders (V3)	**136,426**	1,693,362	91.94%
Tennis (V1)	**130,270**	1,687,206	92.27%
Tutankham (V3)	**131,296**	1,688,232	92.22%

Table 2: Performance comparison

Environment name	Difficulty	Actions	DCapsDQN score ± S.D	Baseline score ± S.D	Performance
Alien (V3)	-	18	**1678.20** ±261	1503.79 ±351	11.60%
Boxing (V1)	-2.11368712	18	**92.87** ±6	58.74 ±18	58.10%
Breakout (V2)	-0.44196066	4	**259.4** ±59	191.1 ±87	35.74%
Fishing Derby (V3)	1.28989165	18	**-11.99** ±14	-27.19 ±14	55.90%
Pong (V1)	-0.04440702	3	**20.15** ±0.8	18.25 ±1.7	10.41%
Qbert (V2)	1.39864132	6	**9942.95** ±1918	5616.26 ±1349	77.03%
Space Invaders (V3)	0.16420283	6	787.64 ±172	**924.11** ±232	-14.76%
Tennis (V1)	10.48605210	18	**-7.138** ±6	-23.645 ±0.98	69.79%
Tutankham (V3)	1.98175005	8	**148.75** ±37	129.20 ±61	15.13%

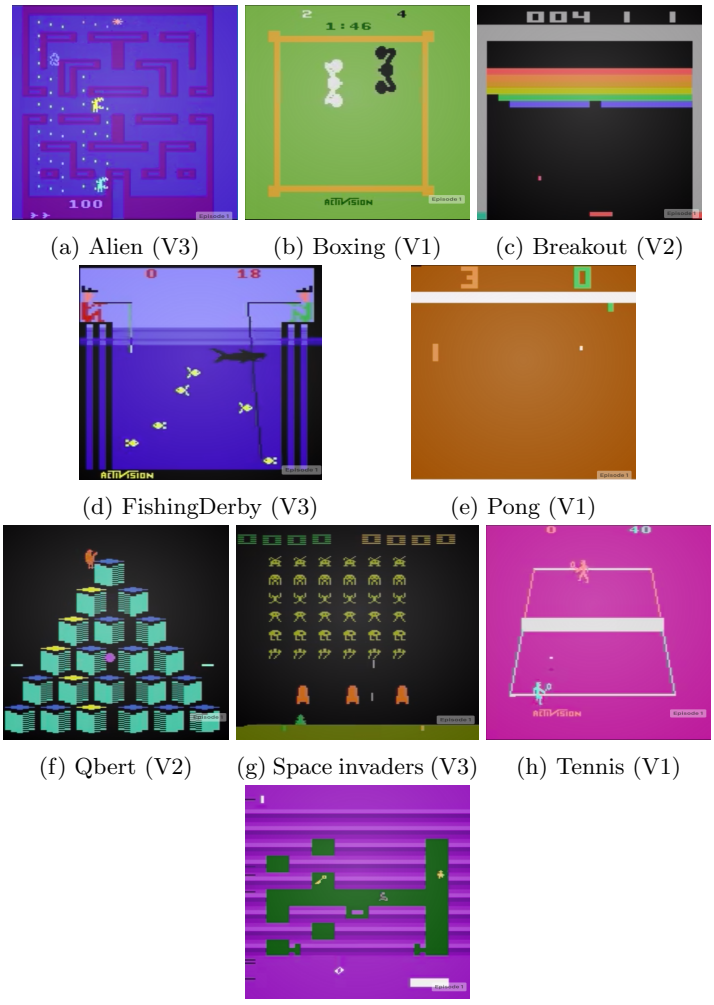
of training parameters, they outperform baseline in all selected environments except SpaceInvaders.

Further in our study, we try to rationalise about the higher cumulative reward collected by DCapsQN on individual attributes of the environment like input state (Sec 5.2), action space (Sec 5.3) and reward (Sec 5.4). We also discuss, how they supplement to cumulative reward in discussion (Sec 6.2).

It is also observable that there is co-relation between difficulty score and average score of DCapsQN. With low difficulty environments like Pong and Boxing, the average score by DCapsQN is more stable and has lower degree of noise compared to the baseline. However with higher difficulty score environment like Tennis or Qbert, we witness a very high standard deviation (S.D) and noisier average score.

5.2 Input state

In this section, we reason how the input state of an environment (Fig. 5) is an influencing factor for DCapsQN agent. The CapsNets architecture focuses on recognising simple and complex entities. As shown in Fig. 5 we can organize the environments in terms of a number of entities and their visual attributes. Pong, Boxing, Tennis are one of the visually simple environments with low number of entities, referring to them as *V1*. Breakout and QBert are more complex than V1, referred to as *V2*. But V2 is simpler compared to Alien, SpaceInvaders, Tutankham and Fishing Derby of *V3*.



(a) Alien (V3) (b) Boxing (V1) (c) Breakout (V2)

(d) FishingDerby (V3) (e) Pong (V1)

(f) Qbert (V2) (g) Space invaders (V3) (h) Tennis (V1)

(i) Tutankham (V3)

Fig. 5: State input of various Atari environments.

It is observable that in the simpler input state of V1, a DCapsQN agent performs excellently. The performance could be highly attributed to the very simple input state. In these environments, there are clear separate entities such as players, ball in the input state. The DCapsQN agent's learning curve is swifter compared to the baseline Double DQN (Fig.6). With comparably complex V2, the convergence of the DCapsQN-based agent is slower yet they outperform the Double DQN based baseline as well. With added visual complexities and an increase in the number of observable objects, we can observe that convergence slows down further. The same can be concluded for V3. The principle that DCapsQN focuses highly on entities further strengthen when comparing the difference in

performance in Tennis(V1) and SpaceInvaders(V3). DCapsQN outperforms the baseline agent which struggles to learn with simpler input state that has clear separate entities in Tennis(V1) (Fig.4h). However DCapsQN struggles where there are multiple copies of the same entities in SpaceInvaders(V3) (Fig.4g).

5.3 Action space

The atari suite provides a variety of environments with respect to action space as well. The action space is an important part of an environment since it is directly related to the number of actions available for the agent. A larger action space expresses a higher degree of freedom for an agent to choose an action from. For our study, we started with a small action space of 3 and 4, in Pong and Breakout, respectively. From there, we go to the largest action space available in atari, i.e 18, in Alien, Boxing, Fishing Derby and Tennis. As can be noticed in Table 1, apart from the expected increase in the number of parameters introduced by the fully connected layers, there does not seem to be a direct correlation between an agent's performance and the action space.

5.4 Reward

In RL, the agent interacts with the environment to get a reward signal and the next state. With the goal of maximising the cumulative rewards, the reward as part of the environment governs how well an agent comprehends the input state. The environments in ALE can broadly be classified into dense rewards or sparse rewards environments. For our investigation, we diversify our environments with some dense reward environments such as Alien and some marginally sparse environments such as Fishing Derby. DQN suffers from poor sample efficiency when rewards are very sparse in an environment [8]. There is a relation between reward density and convergence of an agent to a value function. In the dense reward environment Alien, it takes around 3 million steps for a DCapsQN based agent to outperform the baseline while in Fishing Derby, it takes around 13 million (Fig. 4).

6 Discussion

6.1 Training

DQN [19] based algorithms use their own estimates to update their value. In order to analyze and gain insight into the potential of part-whole relations based representations, we plot and compare the loss (Fig. 7) and value estimates (Fig. 6) of both agents while training.

Fig. 6 compares the value estimates over time from DCapsQN and the baseline. Value functions estimate how good it is to perform a given action in a given state. The notion of "how good" here is defined in terms of future rewards or expected return [27]. A high oscillation of value estimates in consecutive steps

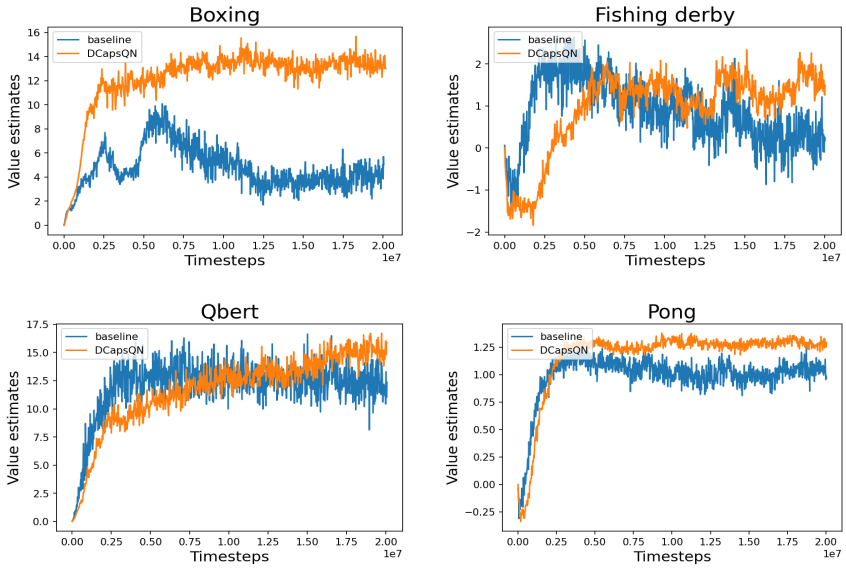


Fig. 6: Value estimates comparison of agents in various environments. It is noticeable that the baseline is more volatile compared to DCapsQN.

translates to a high uncertainty of future rewards. We can observe the difference in magnitude and higher oscillation in consecutive steps between baseline and DCapsQN. We hypothesize that vectored representations in CapsNets further help in stabilizing the change in value function of Double DQN. The hypothesis is further supported by comparing the loss (Fig. 7) of DCapsQN and the baseline. The losses in DCapsQN are comparatively smaller in magnitude compared to those from the baseline agents. This can be attributed to a lower change in weights because the target is often very close to the agent's current estimate. The low magnitude of loss in DCapsQN also indicates that CapsNets do not start representing new entities.

6.2 Environment

While we rationalize the better performance of DCapsQN based agents, there is not a single most powerful component that directly contributes to it. It is the combination of all three elements i.e action space, reward and input state.

It is noticeable the performance of the agent in the environment Tennis is similar to Boxing although they both have a different difficulty level. The leading performance of DCapsQN based agents in both environments can be attributed to very simple visual input and high action space. If compared to the difference in the convergence of agents in Alien (a maze traversal environment and with a highly dense reward) with Tutankham, which is maze traversal but with a comparatively sparse reward environment. We notice that the combination of

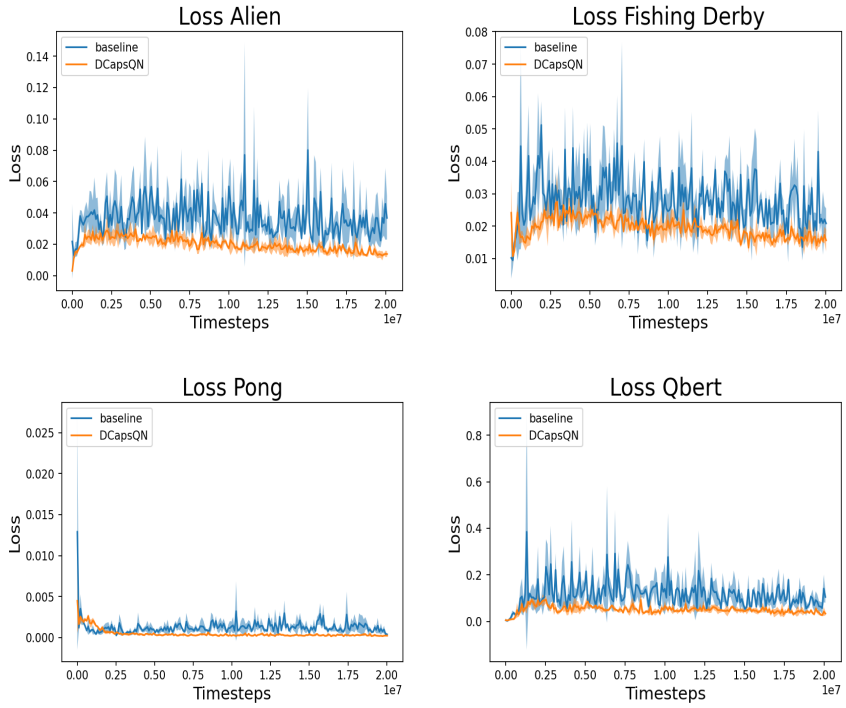


Fig. 7: Training loss comparison of agents in various environments. The shaded area represents the ± standard deviation over 4 runs.

reward and action space contributed more to the performance, compared to the visual input state.

Human perception suffers from crowding, The DCapsQN based agent seems to a show similar phenomenon in SpaceInvaders. The low performance could be attributed to the combination of crowding and low action space, where there are multiple instances of the same part and whole objects in the input state [22, 25].

7 Conclusion

The paper introduced DCapsQN, a CapsNets-based agent for DRL. We empirically show how CapsNets-based architectures perform well with Double DQN. The DCapsQN architecture uses fewer parameters while still outperforming the baseline agent in terms of cumulative reward collected by an agent in a given task. In contrast to previous research [4] where the agent did not converge, DCapsQN converges to find a value function.

The presented architecture was found to be the best performing in terms of design and capabilities in the environments. The outcome confirms the initial

hypothesis that the value function is learned by the fully connected layers while CapsNets learns to better represent input states.

Based on observations made in this work, we consider that transfer learning of representations learned via CapsNets could be an interesting direction for future research. Once learned part-complex objects, the agent would only need to converge to find the value function. Although our evaluation covered a variety of tasks and reward systems, it would be useful to investigate the performance of the agents in other tasks, domains and within other settings like continuous action spaces.

Acknowledgement

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

1. Afshar, P., Heidarian, S., Naderkhani, F., Oikonomou, A., Plataniotis, K.N., Mohammadi, A.: Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images. Pattern Recognition Letters **138**, 638–643 (2020)
2. Afshar, P., Plataniotis, K.N., Mohammadi, A.: Capsule Networks for Brain Tumor Classification based on MRI Images and Course Tumor Boundaries. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1368–1372 (Nov 2019)
3. Allioui, H., Sadgal, M., Elfazziki, A.: Deep MRI Segmentation: A Convolutional Method Applied to Alzheimer Disease Detection. International Journal of Advanced Computer Science and Applications **10**(11) (2019). https://doi.org/10.14569/IJACSA.2019.0101151
4. Andersen, P.A.: Deep Reinforcement Learning using Capsules in Advanced Game Environments. arXiv:1801.09597 [cs, stat] (Jan 2018)
5. Bahadori, M.T.: Spectral Capsule Networks p. 5 (2018), https://openreview.net/forum?id=HJuMvYPaM
6. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The Arcade Learning Environment: An Evaluation Platform for General Agents. Journal of Artificial Intelligence Research **47**, 253–279 (Jun 2013). https://doi.org/10.1613/jair.3912
7. Eck, D.J.: Introduction to Computer Graphics. David J. Eck (2016)
8. Gou, S.Z., Liu, Y.: DQN with model-based exploration: Efficient learning on environments with sparse rewards. arXiv:1903.09295 [cs, stat] (Mar 2019), https://arxiv.org/abs/1903.09295
9. Hinton, G., Sabour, S., Frosst, N.: Matrix capsules with EM routing. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=HJWLfGWRb
10. Huang, W., Zhou, F.: Da-capsnet: dual attention mechanism capsule network. Scientific Reports **10**(1), 1–13 (2020)
11. Hubel, D.H., Wiesel, T.N.: Shape and arrangement of columns in cat's striate cortex. The Journal of Physiology **165**(3), 559–568 (Mar 1963). https://doi.org/10.1113/jphysiol.1963.sp007079

12. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A Convolutional Neural Network for Modelling Sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 655–665. Association for Computational Linguistics, Baltimore, Maryland (2014). https://doi.org/10.3115/v1/P14-1062
13. Kempka, M., Wydmuch, M., Runc, G., Toczek, J., Jaśkowski, W.: Vizdoom: A doom-based ai research platform for visual reinforcement learning. In: 2016 IEEE Conference on Computational Intelligence and Games (CIG). pp. 1–8. IEEE (2016)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Communications of the ACM **60**(6), 84–90 (May 2017). https://doi.org/10.1145/3065386
15. LaLonde, R., Bagci, U.: Capsules for Object Segmentation. arXiv:1804.04241 [cs, stat] (Apr 2018)
16. Liao, H.: Capsnet-tensorflow (2018), https://github.com/naturomics/CapsNet-Tensorflow/blob/master/imgs/capsuleVSneuron.png
17. Martnez-Plumed, F., Hernandez-Orallo, J.: AI results for the Atari 2600 games: Difficulty and discrimination using IRT. In: Evaluating General-Purpose AI. p. 6 (2017)
18. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602 [cs] (Dec 2013)
19. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (Feb 2015). https://doi.org/10.1038/nature14236
20. Molnar, T., Culurciello, E.: Capsule Network Performance with Autonomous Navigation. International Journal of Artificial Intelligence & Applications **11**(1), 1–15 (Jan 2020). https://doi.org/10.5121/ijaia.2020.11101
21. Pan, C., Velipasalar, S.: Pt-capsnet: A novel prediction-tuning capsule network suitable for deeper architectures. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11996–12005 (2021)
22. Pelli, D.G.: Crowding: A cortical constraint on object recognition. Current Opinion in Neurobiology **18**(4), 445–451 (Aug 2008). https://doi.org/10.1016/j.conb.2008.09.008
23. Phaye, S.S.R., Sikka, A., Dhall, A., Bathula, D.: Dense and Diverse Capsule Networks: Making the Capsules Learn Better. arXiv:1805.04001 [cs] (May 2018)
24. Rawlinson, D., Ahmed, A., Kowadlo, G.: Sparse Unsupervised Capsules Generalize Better. arXiv:1804.06094 [cs] (Apr 2018)
25. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 3856–3866. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf
26. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. In: Bengio, Y., LeCun, Y. (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings (2016), http://arxiv.org/abs/1511.05952
27. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)

28. van Hasselt, H., Guez, A., Silver, D.: Deep Reinforcement Learning with Double Q-learning. arXiv:1509.06461 [cs] (Dec 2015)
29. Wen, X., Han, Z., Liu, X., Liu, Y.S.: Point2spatialcapsule: Aggregating features and spatial relationships of local regions on point clouds using spatial-aware capsules. IEEE Transactions on Image Processing **29**, 8855–8869 (2020)
30. Wu, Y., Ma, S., Zhang, D., Sun, J.: 3d capsule hand pose estimation network based on structural relationship information. Symmetry **12**(10) (2020). https://doi.org/10.3390/sym12101636, https://www.mdpi.com/2073-8994/12/10/1636

A Bayesian Framework for Evaluating Evolutionary Art

Augustijn de Boer* ✉[0000−0002−8657−8959], Ron Hommelsheim*[0000−0002−9074−2985], and David Leeftink*[0000−0002−9542−3334]

Radboud University Nijmegen, The Netherlands
{augustijndeboer,ronh93,hdfleeftink}@gmail.com

Abstract. Recent advances in computer-generated art (CGA) have led to a diverse state of generative art models, however, how to evaluate the works produced by these methods remains an open question, due to the subjective nature of the domain. In this work, we propose a framework for evaluating evolutionary art using a Bayesian approach.
The framework provides a method to analyse the results of a number of 'art Turing tests' (ATTs) with a Bayesian model comparison, to assess the influence the evolutionary process has on the degree to which computer-generated images are distinguishable from human generated images.
The cases where the human- and computer-generated art can and can not be distinguished are represented by the null hypothesis and the alternative hypothesis, respectively. We demonstrate the framework using Interactive Evolutionary Computation (IEC) to evolve images with a function-tree representation. These images are then used in an ATT in which $n = 11$ subjects participated. The results indicate a weak preference for the alternative hypothesis, showing that the human- and computer-generated images can not reliably be distinguished. We sketch future applications of the framework, such as evolving cellular automata or combining the framework with deep learning approaches to CGA. The framework is available as an open-source code base, and can be used by researchers and practitioners interested in evaluating their methods for generating evolutionary artworks.

Keywords: Computational Creativity · Evolutionary Computation · Interactive AI Methods and Applications · Bayesian Statistics · Genetic Programming

1 Introduction

Since the infancy of computers, mathematicians, programmers and eventually artists have been intrigued by the new ways in which art could be created. Cellular automata have been used to either create or modify images [11], many different types of fractals can be generated by computers easily [24], developments in deep learning in the last decade has allowed artists to create art, e.g.,

* Equal contributions

by using style transfer [10] and Generative Adversarial Networks [9], genetic algorithms can be utilized to create art by the iterative process of "survival of the fittest" [21], and the list goes on.

Here, we will focus on applying Evolutionary Algorithms for generating art. Evolutionary Algorithms are loosely inspired by the Darwinian theory of evolution by natural selection, which despite of its relative simplicity, describes all life in its enormous complexity. The fittest individuals of a population reproduce often, passing on their genes. The genes mutate and recombine, subsequently producing new individuals. This process has been abstracted and modified many times to solve problems such as parameter estimation or agent-based modeling. It has also been simulated to better understand the actual biological mechanism [18]. Furthermore, the generation of art by EAs has been explored by many artists and researchers alike in a variety of different approaches, this is commonly called Evolutionary Art (EArt) [21].

Section 2 provides a short introduction into evolutionary art, and the way it is currently evaluated. In Section 3, we propose a framework for evaluating EArt using a Bayesian approach. In Section 4 we demonstrate this framework by applying it to a specific case, in which a weak preference for the alternative hypothesis is found. We briefly discuss these results in Section 4.3. The results of a short questionnaire about the experience of working with the framework are discussed in Section 5, and we sketch future directions and applications in Section 7.

2 Background

Loosely inspired by Darwinian evolutionary systems, Evolutionary Algorithms (EAs) can be broken down to a few essential components [2, 12]: an initialization procedure; a fitness function; a selection procedure; a crossover procedure; and a mutation procedure.

The EA cycle starts by initializing a population of individuals. These individuals are all evaluated using the fitness function, after which a number of them is selected. That selection of individuals is then crossed over and mutated to form a new population. This is repeated until some termination criterion is met.

Because the fitness function that is used is unrestricted, EAs allow human feedback as well as computer feedback to be used for evolution. When human feedback is used as a fitness function in EAs, we call this Interactive Evolutionary Computation (IEC). The dependency of the IEC framework on human evaluation as a fitness function is considered one of its core strengths. Nonetheless, the amount of control a user has over the process is still very limited; the selection, crossover, and mutation procedures are governed by pseudo-randomness.

2.1 Evaluating Computer-generated Art

The Turing test (TT) can be used to assess whether a computer is capable of exhibiting (human) intelligent behavior [23, 22]. In the TT, a subject has to

distinguish a human from a computer by only communicating to them through a text channel. If the computer is indistinguishable from the human, it passes the Turing test. Following that line of thought, to assess whether a system is creative, one could devise a Turing test specifically for art, or an 'Art Turing test' (ATT), as introduced by Boden [3]. In an ATT, a subject has to evaluate two pieces of art, one created by a computer, and one created by a human, and decide which one of them was created by a human. This way of evaluating art may seem fair at first, but Pease et al. [19] pose some objections. Mainly, their point is that the ATT does not allow the subject to interact with the art, as opposed to the classical TT, where the subject can interact with the human and the computer. Much information about the art that could influence the subject can not be taken into account this way. Similarly, the ATT does not take into account framing information. Another argument they pose is that the ATT encourages imitation, and not creativity. Lamb et al. [16] do acknowledge that the ATT is only valid in those cases where the CGA is specifically designed to imitate human art. In this paper we use an ATT to compare CGA and human art that were both made with the same method, thereby satisfying the constraints set by Lamb et al. To the best of our knowledge, we are the first to evaluate evolutionary art using the methods described here.

3 The Bayesian Framework

We propose to evaluate evolutionary art by doing a Bayesian Model Comparison (BMC) on results from an art Turing test. Here we provide an explanation of the framework and the methods used, as well as a .

3.1 Art Turing Test

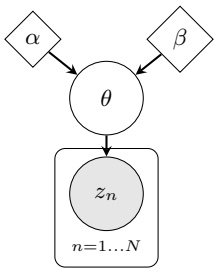
We use an ATT to determine whether the evolutionary process has an influence on the degree to which the human generated images can be distinguished from computer-generated images by humans. To this end, three pools of images need to be generated by EAs.

- One pool is generated by letting a human act as a fitness function for multiple sessions of 10 generations. After the 10th generation, all images in the population are added to the pool of so-called 'human-generated' images. Note that although these images are called 'human-generated', the influence the human has on the generative process is limited. Images from this pool are indicated with h_{10}.
- One pool of computer generated images is created in the same way, but instead of using the human evaluation, we use an automatic fitness function. Images from this pool are indicated with c_{10}.
- The other pool of computer-generated images is created from purely random initial trees, i.e., they are evolved to generation 1. Images from this pool are indicated with c_1.

During the ATT, the user is to decide which of two presented images is human-generated. The pair of presented images can be one of two possible combinations, either a (h_{10}, c_{10}) pair, or a (h_{10}, c_1) pair, both being equally likely. The subject does not know which is being presented, and is not aware that this difference between the cases exists. Naturally, the images within a pair are randomly ordered when they are presented to the user.

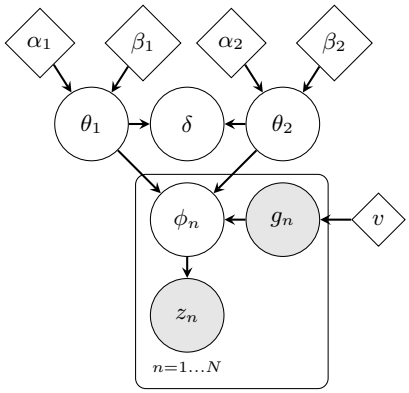
3.2 Bayesian Model Comparison

Null Hypothesis $\mathbf{H_0}$ The probability with which a participant answers correctly on the Turing test is fixed and does not depend on whether the decision was on a (h_{10}, c_1) pair or a (h_{10}, c_{10}) pair. We let z_i be 1 if the answer on the i'th Turing test was correct, and 0 if it was incorrect. We can express this in a graphical model $\mathcal{M}_0$, as shown in Figure 1:



$$\theta \mid \alpha, \beta \sim \text{Beta}(\alpha, \beta)$$
$$z_n \mid \theta \sim \text{Bernoulli}(\theta)$$

Fig. 1: Graphical model $\mathcal{M}_0$ for H_0



$$\alpha_i, \beta_i = 1, 1$$
$$\theta_i \mid \alpha_i, \beta_i \sim \text{Beta}(\alpha_i, \beta_i)$$
$$\delta \mid \theta_1, \theta_2 = \theta_1 - \theta_2$$
$$v = 0.5$$
$$g_n \mid v \sim \text{Bernoulli}(v)$$
$$\phi_n \mid \theta_1, \theta_2, g_n = \begin{cases} \theta_1 & \text{if } g_n = 1 \\ \theta_2 & \text{if } g_n = 10 \end{cases}$$
$$z_n \mid \phi_n \sim \text{Bernoulli}(\phi_n)$$

Fig. 2: Graphical model $\mathcal{M}_1$ for H_1

Alternative Hypothesis $\mathbf{H_1}$ The probability with which a participant answers correctly on the Turing test depends on whether the decision was on a

(h_{10}, c_1) pair or (h_{10}, c_{10}) pair. The graphical model corresponding to this hypothesis can be found in Figure 2. The variables θ_1 and θ_2 are used for each of the two possible pairs of images. In this graphical representation, if the $i'th$ decision was made on a (h_{10}, c_j) pair, we indicate that with $g_i = j$.

The Bayes Factor The Bayes Factor (BF) was used as a measure to compare models $\mathcal{M}_0$ and $\mathcal{M}_1$. The BF is the ratio of the marginal likelihoods of the two models: $B_{10} = \frac{p(Z|\mathcal{M}_1)}{p(Z|\mathcal{M}_0)}$. To estimate the BF one can construct an hierarchical Bayesian model in which the selection for model $\mathcal{M}_0$ or $\mathcal{M}_1$ is part of the sampling process, and governed by a categorical distribution. The ratio of the frequency that each model was selected can be used as an estimate for the BF. The BF acquired this way is then interpreted according to, for example, Kass et al. [13]. Furthermore, the variable δ expresses the difference between the two cases in $\mathcal{M}_1$, in terms of how easy it was to distinguish the h_{10} images from the c_1 or c_{10} images.

4 Application

In this section we will apply our framework to a case where images are evolved by Genetic Programming. First we will provide an explanation of Genetic Programming and the type of representation that was used, then we discuss the fitness function that we propose to generate art by mimicking human evaluation. Lastly, we will analyze the results of the ATTs and briefly discuss those results.

4.1 Tree Representation

A Genetic Algorithm (GA) is a type of EA where a distinction is made between the genotype and phenotype of an individual [12]. The genotype represents the underlying structure by which a potential solution is represented. Commonly used representations for the genotype are character strings, trees, or real-valued vectors. The phenotype represents the physical traits of individuals. This distinction is central to the field of evolutionary computation, as it allows for dynamical change of the population via cross-over and mutation between genotypes of the population members. Genetic Programming (GP) [15] is a specific type of GA where the phenotype is a computer program, or—as in our demonstration—a mathematical function.

Whereas EAs such as GA and Evolution Strategies (ES) commonly use linear structures (such as bit strings and real-valued vectors) for the genotype, one can alternatively construct a non-linear genotype using a tree representation [12, 1]. In this demonstration, the genotype is a tree representation (TR), and the phenotype is a mathematical function, which is applied to a grid of pixels to generate an RGB image. Here one could say that the generated image is a plot

of the phenotype, or that the image is the phenotype itself. A TR is a recursive structure consisting of terminal and non-terminal nodes. Terminal nodes are either variables or constants, whereas non-terminal nodes are n-ary functions.

Crossover between two trees happens with probability p_c, by exchanging a random node in the first tree with a random node in the second tree. The children of the exchanged nodes are also moved to the other tree, so we call it a transplantation. Mutation in trees normally happens with probability p_m, by randomly changing the function of a function node, or replacing a leaf node with a new structure. We found that we already achieved pleasing results without mutation, and in literature it is stated that very low mutation rates are suitable for trees [14], so we decided to not apply mutation. In our tree evolution runs, we always set p_m to 0.

We experimented with several versions of tree representations to create RGB images. Our first representation maps every point in a 2D grid to a single numeric value, and then maps each numeric value to a RGB value using a color gradient. Our second representation creates a separate tree for each 2D color channel, and normalizes each layer separately to lay within the correct interval $[0, 255]$. These layers are then stacked to create an RGB image. Our third representation is a single tree which can map 3D coordinates to a numeric value. Like the second version, the color channels are normalized individually.

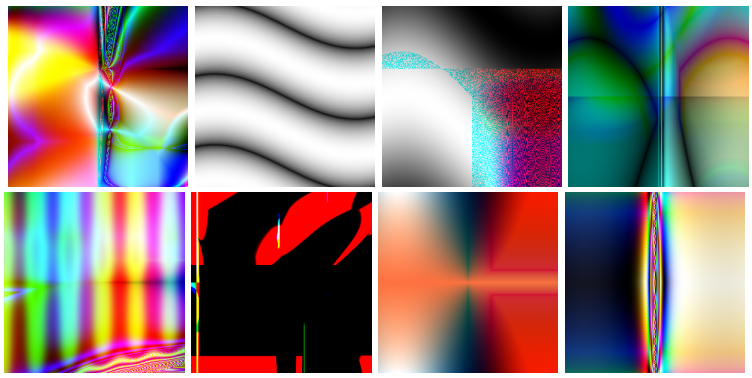


Fig. 3: Examples of tree representation-based images from c_{10}

An excellent illustrated overview of the crossover and mutation methods in these tree representations can be found on Ashley Mills' website [1].

4.2 The Mathematical Fitness Function

In this section we present ideas that went into designing the fitness function that the computer uses to evaluate the images will be presented. We hypothesise that if presented with a small population—say a population consisting of 9

individuals—a person would evaluate the individuals by the characteristics that make them stand out from the other individuals in the population. Following this line of thought, one could define a human-inspired mathematical fitness function F for an individual p, as the mean distance of individual p to each of the other individuals in the population P:

$$F_D(p) = \frac{1}{|P| - 1} \sum_{p' \in P, p' \neq p} D(p, p') \tag{1}$$

Any distance metric can be used, for example the Euclidean distance. Using the Euclidean distance does not yield very interesting images, however. Suppose in a population we have one entirely white image, and one entirely black image. The Euclidean distance if evaluated in the RGB space is maximized, since the RGB components of white are (255,255,255) and the RGB components of black are (0,0,0). As a result, these images will be assigned a high fitness, even though they are (subjectively) very uninteresting. A more interesting approach would be to use the variance of the difference of the pixel values as a distance function. Using the example of the entirely white and entirely black image again, the distance between these two images will now be 0; the difference between every pair of white and black pixels is the same. This approach yielded more interesting images, see equation 2.

$$D_{\mathrm{Var}}(p, p') = \mathrm{Var}(p - p') \tag{2}$$

The pool of c_{10} individuals used in the experiment was evolved using the function described in Equation 1 with the distance measure from Equation 2 as a fitness function. The pool of c_1 individuals was generated by simply randomly initializing trees. The pool of h_{10} individuals was evolved by letting a human act as the fitness function by rating the images produced by them. Starting from the root node, working downward, each node is uniformly sampled from either the binary or unary functions, or the leaf nodes. Within each category, the specific selection is again sampled uniformly from $\{+, -, \times, \div, \mathrm{power}, \min, \max\}$, $\{\sin, \cos, \tan, \mathrm{abs}, \sqrt{}\}$, and $\{x, y, z, 0.618\}$, respectively.

4.3 Results and Analysis

The h_{10} pool used here was generated by the authors, who do not have a formal art education. The experiment was done with $n = 11$ participants, each of which performed 20 ATTs, resulting in 220 binary (correct/incorrect) results. The age of the participants ranged between 20 and 27, and none of them had a formal art education. The average interaction time per participant was around 15 minutes. Of the 220 ATTs, 97 were answered correctly, about 44%. The results of the ATTs on the sub-classes are listed in Table 1.

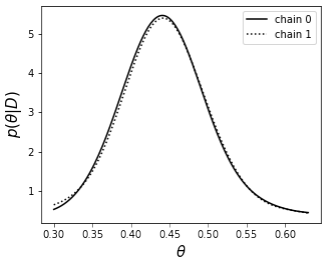
Table 1: Results of the ATTs

	All	c_1	c_{10}
Total	220	94	126
# Correct	97	38	59
% Correct	44%	40%	47%

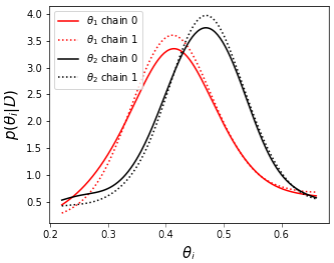
Table 2: Relative sampling frequencies f_s for each model

Model	rank	f_s
$\mathcal{M}_0$	1	0.468
$\mathcal{M}_1$	0	0.532

After running two Markov chains of 5000 samples for each model, our samplers over the model parameters converged nicely to some interesting distributions, which can be seen in Figure 4. It is interesting to see that θ_1 peaks at a lower value than θ_2. This seems to imply that participants have a lower chance of answering the ATT correctly if the computer generated image is completely random, and not evolved using the automatic fitness function.



(a) θ, governing $\mathcal{M}_0$, was estimated to have a mode below 0.5.



(b) θ_1 was estimated to have a lower mode than θ_2.

Fig. 4: Density estimates

It would be premature to say that the use of the automatic fitness function actually makes the art look *less* human-like, but that is what the numbers seem to indicate. Still, the peak of θ_2 is also lower than 0.5, meaning that the human-generated images are often correctly identified.

Model $\mathcal{M}_0$ was sampled in 46.8% of the cases during the BMC. Model $\mathcal{M}_1$ was sampled in the remaining 53.2% of the cases (see Table 2), resulting in an estimated Bayes factor of 1.14. According to Kass et al. [13], this is weak support for the alternative hypothesis.

Although the BMC showed weak preference for the alternative hypothesis, there is too little evidence to reject H_0. We can not conclude that images generated by function trees evolved using the automatic fitness function are perceived as more human-like than images generated by random function trees. However, the number of participants in our experiment was small, and with more participants it may be possible to give a more conclusive answer.

5 Questionnaire

All participants were asked to fill in a questionnaire after interacting with the evolutionary framework through the GUI. The questions and the results of that questionnaire are listed in Table 3.

Table 3: Results of the questionnaire, entries are counts

	Strongly disagree						**Strongly agree**
1.I enjoyed the process of making images interactively.	0	0	1	1	2	3	4
2.I have the feeling that the image is improving with increasing number of iterations.	0	0	1	3	2	1	4
3.I feel that I have control over the evolution of the images.	0	1	0	3	3	2	2
4.The generated images were surprising to me.	0	1	0	2	2	3	3
5.I find the generated images pleasant.	0	1	0	2	4	3	1
6.I want to know how the underlying mechanism works.	0	1	1	0	3	0	6

The quality and responsiveness of the evolutionary process is rated positively in general, but indicate that there is still room for improvements. Question 4 addresses the extent to which participants felt control over the evolution of the art, which resulted in a mode of 4 and 5, a median of 5 and a mean of 5. This was a positive outcome, with one outlier on the lower end. Question 5 covers the degree of surprise of the images, and was perceived positively with a mode of 6 and 7, a median of 6 and a mean of 5.36. Again, we find one low outlier with a rating of 2. Lastly, question 6 addresses the degree to which participants found the images pleasant. The results indicate a mode of 5, a median of 5 and a mean of 5.55. Again, we find one negative outlier at 2.

Based on the questionnaire results, we conclude that the IEC framework is perceived very positively. Participants generally enjoy the process of creating images and are curious about the underlying mechanisms. Furthermore, participants notice the improvement of images as a function of generations. The results also indicate room for improvement when it comes to the quality of the generated art. In particular, participants showed lower scores for control over the generated art. We hypothesize that this is related to the relative small population size (a population size of 9 is used at each iteration), which can make the process susceptible to losing the fittest individuals in the population due to the stochasticity of the crossover function. Lastly, we conclude that even though the tool is generally highly perceived, outliers exist, which indicates that there are strong differences between participants in how the application was used and perceived.

6 Code Base

A primary result of this project is an open-source code base written in Python which includes many variations of the basic components of evolutionary algorithms listed in Section 1, and which can be easily extended to include more. This Python code also includes a GUI that allows the user to perform the interactive evolution, and to perform the ATTs required for the proposed framework. The project can be found on GitHub [4].

7 Discussion

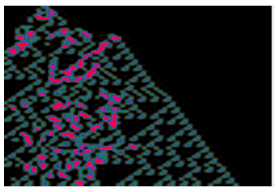
The demonstration of the ATT using a function-tree representation showed that participants scored worse than chance, meaning computer generated art could not reliably be distinguished from human generated art created with the IEC framework. We hypothesize that this could be caused by the lack of control of the creative process that is given to participants while using the function-tree representations. This is in line with the questionnaire results, which highlight that the evolved images using function-tree representations were generally perceived well by the participants, but the control over the evolutionary process can still be improved. We hypothesize that the choice of selection strategy can be of influence on this: by using roulette-wheel selection, individuals with high ratings are likely to stay in the population. This however also quickly filters out images with low ratings, causing the process to converge faster than desired. In contrast, different selection mechanisms such as tournament selection can cause good solutions to disappear despite high ratings, but retains solutions with low ratings better than roulette-wheel selection.

We propose several directions for future research, which may provide further improvements to the statistical framework, and the code base.

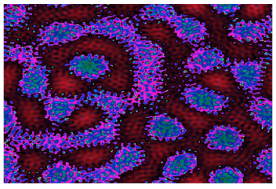
First, we believe that the use of different selection mechanisms such as steady state selection [20] and Boltzmann selection [17], or techniques like elitism [12] may improve the control of participants over the evolutionary process.

Second, the set of functions that are used to construct the function-tree can be extended. Since these directly influence the images, this can have a significant effect on their ratings. Moreover, extensions to our work could include different representations. We ourselves have experimented representing individuals as Cellular Automata (CA), such as in Conway's "Game of Life" [8]. We extended these CAs by generalizing the discrete states to intervals and the discrete time domain to acceleration, such as in Chan's "Lenia" [6, 7]. Results of both representations can be seen in Figure 5a and 5b, respectively.

Third, the presented framework is readily extendable to be applied to different types of evolutionary art representations, such as representations based on deep learning. For example, Bontrager et. al (2018) [5] combine Generative Adversarial Networks (GANs) and IEC to evolve images. Applying different representations of artworks in the presented framework is a promising direction of future research.



(a) Cellular automaton



(b) Multi-neighborhood cellular automaton

Fourth, the many potential uses of the framework can be exploited; for instance, one could study the influence of the evolutionary process on the perceived creativity of the process underlying the art generation with a finer granularity than was done here. In our demonstration, we generated pools of c_1 and c_{10} images, but one could easily extend that to include c_n images, and compare the influence of the generation depth on the Bayes Factor. Additionally, one could use the framework as a competition between several types of evolutionary art. Lastly, the ATT could be interpreted more freely, and instead of asking the subject which of the presented images was perceived to be more likely to be generated by a human, one could ask the subject simply which of the images he/she liked more. In a world where computer-generated art is ubiquitous, a flexible statistical framework like this may prove a valuable tool.

The questionnaire results showed that the application was found very enjoyable and quite intuitive, which is why we believe extending the framework and the code base is a venture worth pursuing.

8 Conclusion

Art is subjective. Nonetheless, complex and often interesting patterns can emerge using the techniques of algorithmic evolution. Utilizing the input of users in an Art Turing Test, we frame the task of evaluating generated art as the degree to which computer generated art can be distinguished from human generated art. Using a Bayesian model comparison, we created a framework for inferring whether the difference in degree of distinguishability is significant. The proposed automated fitness function scored worse than non-evolved function-trees in the ATT, although the results are inconclusive. We conclude that this means the method can be further improved to provide more control over the evolutionary process of generating images. We provide a framework for IEC using function-tree and CA representations, which allow the user to provide feedback on the generated individuals. The framework is open source and easily extendable to different representations, allowing for researchers and practitioners to adopt it efficiently. Results from an experiment show that the method is well-perceived in general, however improvements can still be made to the representations.

References

1. Evoart - Ashley Mills. https://www.ashleymills.com/art/evoart/
2. Back, T.: Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford university press (1996)
3. Boden, M.A.: The Turing test and artistic creativity. Kybernetes (2010)
4. de Boer, A., Leeftink, D., Hommelsheim, R.: Augub/evaluating$_e$volutionary$_a$rt(Oct2021), $https$: $//github.com/AuguB/evaluating_evolutionary_art$
5. Bontrager, P., Lin, W., Togelius, J., Risi, S.: Deep interactive evolution. In: International Conference on Computational Intelligence in Music, Sound, Art and Design. pp. 267–282. Springer (2018)
6. Chan, B.W.C.: Lenia-biology of artificial life. arXiv preprint arXiv:1812.05433 (2018)
7. Chan, B.W.C.: Lenia and expanded universe. In: Artificial Life Conference Proceedings. pp. 221–229. MIT Press (2020)
8. Conway, J.: The game of life. Scientific American **223**(4), 4 (1970)
9. Elgammal, A., Liu, B., Elhoseiny, M., Mazzone, M.: Can: Creative adversarial networks, generating "art" by learning about styles and deviating from style norms. arXiv preprint arXiv:1706.07068 (2017)
10. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
11. Greenfield, G.R.: Minimalist art from cellular automata. Journal of Mathematics and the Arts **14**(1-2), 63–65 (2020)
12. Jong, K.D.: Evolutionary computation. Wiley Interdisciplinary Reviews: Computational Statistics **1**(1), 52–56 (2009)
13. Kass, R.E., Raftery, A.E.: Bayes factors. Journal of the american statistical association **90**(430), 773–795 (1995)
14. Koza, J.R.: Evolution of subsumption using genetic programming. In: Proceedings of the First European Conference on Artificial Life. pp. 110–119 (1992)
15. Koza, J.R.: Genetic programming: on the programming of computers by means of natural selection, vol. 1. MIT press (1992)
16. Lamb, C., Brown, D.G., Clarke, C.L.: Evaluating computational creativity: An interdisciplinary tutorial. ACM Computing Surveys (CSUR) **51**(2), 1–34 (2018)
17. Lee, C.Y.: Entropy-Boltzmann selection in the genetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **33**(1), 138–149 (2003)
18. Messer, P.W.: Slim: simulating evolution with selection and linkage. Genetics **194**(4), 1037–1039 (2013)
19. Pease, A., Colton, S.: On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. In: Proceedings of the AISB symposium on AI and Philosophy. vol. 39 (2011)
20. Rogers, A., Prügel-Bennett, A.: Modelling the dynamics of a steady-state genetic algorithm. Foundations of genetic algorithms **5**, 57–68 (1999)
21. Romero, J., Machado, P.: The art of artificial evolution: A handbook on evolutionary art and music. Springer Science & Business Media (2008)
22. Turing, A.M.: Computing machinery and intelligence. In: Parsing the Turing test, pp. 23–65. Springer (2009)
23. Turing, A.M.: Mind. Mind **59**(236), 433–460 (1950)
24. Xujian, Q.: Fractal art. Journal of Jilin College of the Arts p. 05 (2007)

Exploring Explainable AI in the Financial Sector: Perspectives of Banks and Supervisory Authorities

Ouren Kuiper[1[0000-0002-5033-6173]], Martin van den Berg[1[0000-0003-3974-7374]], Joost van der Burgt[2], and Stefan Leijnen[1]

[1] HU University of Applied Sciences Utrecht, Utrecht, Netherlands
{ouren.kuiper, martin.m.vandenberg, stefan.leijnen)@hu.nl
[2] De Nederlandsche Bank, Amsterdam, Netherlands

Abstract. Explainable artificial intelligence (xAI) is seen as a solution to making AI systems less of a "black box". It is essential to ensure transparency, fairness, and accountability – which are especially paramount in the financial sector. The aim of this study was a preliminary investigation of the perspectives of supervisory authorities and regulated entities regarding the application of xAI in the financial sector. Three use cases (consumer credit, credit risk, and anti-money laundering) were examined using semi-structured interviews at three banks and two supervisory authorities in the Netherlands. We found that for the investigated use cases a disparity exists between supervisory authorities and banks regarding the desired scope of explainability of AI systems. We argue that the financial sector could benefit from clear differentiation between technical AI (model) explainability requirements and explainability requirements of the broader AI system in relation to applicable laws and regulations.

Keywords: Explainable AI, Artificial Intelligence, Financial Sector.

1 Introduction

In recent years increasingly powerful, but often also increasingly complex, machine learning methods have become available and are used to greater extent in commercial contexts [1,2]. Generally, this form of machine learning is referred to simply as "artificial intelligence" (AI). The increasing use of novel and hard-to-understand types of AI systems has sparked a discussion on the need for explainability of AI [3,4]. Especially for high-risk use cases there is a realization, both scientifically and societal, that AI needs to be explainable to be understood. For instance, the upcoming EU legislature on AI [5] will require demonstrable transparency for which explainable AI will be essential. In the financial sector comprehensive understanding of the use of AI systems is even more crucial: both stipulated by a wide range of laws and regulations and because trust in financial institutions is of high importance [6]. Simultaneously, expectations of new AI systems are high in the financial sector, while regulators need time to keep up with the speed of development [7]. Striking the right balance between performance and explainability can present a difficult dilemma for financial institutions.

2

The field of explainable AI (or 'xAI') studies how AI can be made explainable by making algorithms and their systems more transparent, often referred to as "opening the black box" [3]. An improved understanding of the working of these algorithms helps us to verify them, improve them, and implement them ethically. Most developments in xAI focus on either technical tools for model developers or approach explainability as a social or cognitive challenge [8,9]. Other authors have stated that making models explainable should be foregone instead of using inherently interpretable models [10]. Given the attention transparency and explainability receive as a requirement for ethical AI, it is no surprise that many reports on the responsible use of AI have stressed the need for xAI [11]. Notably, the number of empirical studies that provide practical insights into how xAI is actually used in practice is very limited [12] which we believe represents a hiatus in the current literature.

Financial institutions, both large and SME, have begun to use AI, for instance in delivering instant responses to credit applications, claim settlement, and transaction monitoring [24,25]. The World Economic Forum [16] notes that the opacity of AI systems poses a serious risk to the use of AI in the financial sector: lack of transparency can lead to loss of control by financial institutions and thereby damage consumer confidence and society. Given the crucial role of trust in the financial sector, explainability of the outcomes and functioning of AI systems is considered necessary [16]. Explainability is in fact one of the EU's key requirements for trustworthy AI [11]. With new EU AI legislation announced, explainability is expected to become even more important and necessary for some high-risk use cases such as consumer credit scoring [5].

Limited empirical descriptions on the challenges surrounding the application of xAI exist. In addition, only preliminary guidelines exist [17] on how to implement xAI, often based in theory and lacking empirical validation. In the future, a solid and practical framework could help organizations to better understand their obligations (regulatory and otherwise) regarding xAI and how to operationalize them. In the financial sector, such a framework could also help supervisory authorities to translate current regulations regarding transparency and the provision of information, to clear expectations regarding xAI to regulated entities. In lieu of such a framework, a starting point is to map what is currently expected of in terms of explainability of AI by banks and supervisory authorities.

The current exploratory study aims to identify what the differences are regarding the expectations of explainability of AI for supervisory authorities and regulated entities in the financial sector. Three use cases were examined in which AI is used at financial institutions in the Netherlands. Data were collected by means of semi-structured interviews with interviewees of both banks and supervisory authorities. This study is intended to add empirical data on how xAI is regarded and used in practice and as stepping stone towards a framework as described above. The main research question is: *What are the perspectives of supervisory authorities and regulated entities regarding the application of xAI in the financial sector?*

2 Theoretical background

Explainable AI (xAI), also referred to as interpretable or understandable AI, aims to solve the "black box" problem in AI [18,19]. A typical present-day AI system utilizes data (e.g., information on a person's financial situation) and produces an outcome (e.g., a risk of default indication). However, in such a system it is not always evident from the output how or why a certain outcome is reached based on the data. Especially when using more complex AI systems (e.g., using deep learning or random forest methods) the process from input to output is practically impossible to understand by humans even with full knowledge of the inner workings, weightings, and biases of the system. The term xAI encompasses a wide range of solutions that explain why or how an AI system arrives at outcomes or decisions [20]. One line of research focuses on technical tools to explore the relation between model input and output, such as SHAP [21] and LIME [22]. A critique on the xAI field expressed by various authors is that xAI is often not clearly defined and discussed without proper understanding of the surrounding concepts and the parties involved [19,23]. As such, the exact scope of xAI is not always well-defined, as sometimes the term is used to focus on technical solutions directly relating to the model, but sometimes the system context is also taken into account.

Transparency is one of the central concepts of xAI. Importantly, the term is used in two distinguishable contexts or manners in the literature, which we differentiate by using *model transparency* and *process transparency*. Model transparency is the property of a model to be understood by a human as it is, in terms of its general working or design. The opposite of "black-boxness" is model transparency [3,10]. This type of transparency is generally what model developers refer to and is highly related to the concept of interpretability [24,18]. Process transparency is transparency of the use and development of an AI system; it relates to openness and not concealing information for stakeholders [24]. This form of transparency is generally what the colloquial meaning of transparency refers to. However, it is also the type of transparency that is meant in some of the literature on responsible use of AI when talking about "transparency" [10,17].

Explainability means that an explanation of the operation and outcome of a system can be formulated in such a way that it can be sufficiently understood by the stakeholder [3]. The term "stakeholder" refers to the individual, party, or audience impacted by the functioning and/or outcomes of the AI system, requiring information in the form of an explanation. In a vacuum, i.e., without a stakeholder, an explanation cannot be said to do what is intended, namely making something understood by an individual [9]. We would argue that the core concept of explainable AI is *effectual* explanation. An effectual explanation is not only about providing the required information, but to do so in a manner that leads to stakeholder understanding [25], for instance by offering the right amount of detail or boundary conditions of a model [26]. In addition, explanations can be global or local [13,14,26]. That is, a global explanation reveals the inner workings of the entire AI system (potentially including a case at hand), a local explanation offers insight in a specific outcome.

We used the following definition of explainable AI in this study: "*Given a stakeholder, xAI is a set of capabilities that produces an explanation (in the form of details,*

reasons, or underlying causes) to make the functioning and/or results of an AI system sufficiently clear so that it is understandable to the stakeholder and addresses the stakeholder's concerns." [15].

Various types of information that can be used as the basis for an explanation can be distinguished. A distinction that should be noted here is that of the of process-based versus outcome-based explanation [17]. A process-based explanation gives information on the governance of the AI system across its design and deployment; the explanation is about "the how". An outcome-based explanation gives information on what happened in the case of a particular decision; the explanation is about "the what". In addition, explanations can be said to be "global" (explaining the entire model) or "local" (explaining a specific outcome) [13,14,26]. Furthermore, xAI techniques to gain more information about the functioning of a model can be model-agnostic (and work on any model, e.g., SHAP [21]), or be model-specific.

As a basis for this study we established a list of types of information that can underpin an explanation (of an AI system) that are relevant in the financial sector. We based this list on literature on explainable AI (using snowball search and focusing on the most cited papers in the field) and adapted it to fit use cases in the financial sector ([9,13,14,17,26]) . It should be noted that we incorporated types of information that are related to process-based explanation (e.g. the process surrounding the AI system), and which might be omitted in some views of explainable AI, that are however relevant from a regulatory perspective on AI in finance.

- The reasons, details, or underlying causes of a particular outcome, both from a local and global perspective.
- The data and features used as input to determine a particular outcome, both from a local and global perspective.
- The data used to train and test the AI system.
- The performance and accuracy of the AI system.
- The principles, rules, and guidelines used to design and develop the AI system.
- The process that was used to design, develop, and test the AI system (considering aspects like compliance, fairness, privacy, performance, safety, and impact).
- The process of how feedback is processed.
- The process of how explainers are trained.
- The persons involved in design, development, and implementation of the AI system.
- The persons accountable for development and use of the AI system.

3 Research method

3.1 Use cases

To address our research question, we applied a qualitative research approach by means of a series of semi-structured interviews. Three types of use cases were examined. The two supervisory authorities took part in all three use cases, with each of the three banks partaking in two of the three use cases (due to constraints in availability of

interviewees). The three use cases were: 1) consumer credit, 2) credit risk management, and 3) anti-money laundering. A brief outline of these use cases will now be given.

The use case on consumer credit considers a typical case for consumer credit and a mortgage lending case. Consumer credit is credit provided to a consumer, which can be used to purchase goods and services. Financial institutions that provide consumer credit in the Netherlands have the right and obligation to ensure that the borrower has the capacity to repay the loan. Credit risk management focusses on internal risk and/or capital requirement models (early warning systems and probability of defaults models) where AI systems can be used to improve or replace the currently used models. The use case on anti-money laundering (AML) concerned AI systems which are used to conduct suspicious activity monitoring and transaction monitoring.

3.2 Data collection

The organizations involved in this study are two supervisory authorities (SAs) and three banks in the Netherlands. For reasons of anonymity these will be referred to as "SA", or "first SA", "second SA", "first bank", etc. depending on which interview took place first. The three banks belong to the major banks in the Netherlands, each with more than one million clients, and can be characterized as financial incumbents [27]. Semi-structured interviews were conducted with employees of these five organizations regarding the three use cases. For all interviews, use case experts (i.e., individuals that worked primarily on the use case at hand) were present. These experts either had a technical expertise (those directly involved with the development of the AI system) and/or a more supervising/governing role (such as compliance & risk officers and model owners).

At each interview at least two interviewees of that organization were present, and at most four (if the complexity of the use case required more diverse expertise in the interviewees). Interviews took between 1 and 1.5 hours. In total 13 interviews took place, six with interviewees from supervisory authorities and seven with interviewees from banks (as one bank took part in an additional interview to fully cover all questions). In addition, the findings were refined in a plenary session in which at least one participant of all five organizations was present. As a starting point during the interviews, a list of questions was used to guide the discussion, but the conversation was permitted to develop naturally in the direction deemed most suitable by the interviewers and interviewees.

The interviews with the banks and supervisory authorities had a slightly different list of starting questions, as the SA interviewees did not have the same direct knowledge of a specific use case in contrast to the banks. The interviewees of the banks were asked questions about the following topics: the context of how AI is being used in the organization, the role of explainability in the AI development process, the workings of the use case at hand, the application of AI in the use case, the relevant stakeholders, and how the bank deals with explainability in this particular use case. Finally, the banks were asked what types of information that can serve as a basis for explanations (based on the list from section 2) are considered relevant for supervisory authorities.

For the supervisory authorities, the focus of the interviews was on the boundaries of what they would allow in terms of AI and what their expectations of explainability were for that use case. The interviewees were asked questions concerning: their perception of the use of AI and xAI, applicable legislation around the use case, and the requirements for explainability from a supervisory perspective. In addition, they were asked what types of information (based on the list from section 2) they consider relevant for their supervisory role for the use case at hand. The interviews with the two supervisory authorities were conducted with interviewees who were aware of the applicable prudential, integrity and conduct regulations relating to the use cases.

All interviews were conducted by two researchers of the HU University of Applied Sciences via Webex. During every interview, one of the researchers had the lead in asking questions while the other made notes used for later analysis. After the interviews, the interviewees verified the interview reports and supplemented information where needed.

3.3 Data analysis

Data analysis was conducted based on the interview reports. As a first step we analyzed the interview reports and created a list of the main findings and conclusions per interview. These findings and conclusions were verified and supplemented by the interviewees. As a next step, we analyzed all interview reports and developed an overview of the main conclusions. These conclusions were discussed in a plenary session with participants of the supervisory authorities and banks. The output of this session was used to refine the conclusions.

4 Results

First, we discuss the most notable results per use case. Second, we discuss the overall findings.

4.1 Consumer credit

The first bank provided a use case about mortgage lending (a type of consumer credit) in which an AI system was used to assess mortgages with traffic-light colors to support middle office employees. The AI system runs in parallel to other, more traditional systems in the mortgage approval and monitoring process (e.g. using business rules). The AI system uses a rather simple form of machine learning based on logistic regression and uses around 10 variables. It improves on a business rules system in that it uses historical data. Interestingly, relating to explainability the primary users of the AI system (the middle office employees) were by design not given detailed insight into the functioning and results of the AI system to prevent potential gaming of the system. Due to the relative interpretability of the model, explainability to other stakeholders was not considered to be a challenge beyond the previous systems.

The second bank also supplemented their traditional loan approval system for consumer credit with an AI system. The traditional system uses basic data, such as the data a client provides through the application process or data from credit bureaus. The new AI system is trained and continuously supplied with new transactional data. The combination of both models resulted in fewer defaults on loans. For this use-case, model developers are considered the most important stakeholders regarding explainability. It was stated that it would be possible from a technological point of view to explain the model to customers, although this requires a thorough understanding of which type of narratives would be comprehensible by different consumer groups. This might require an interactive process, which was indicated to present a challenging IT problem rather than a problem of getting the relevant information (and explanations) from the AI system.

One of the SAs monitors whether lenders (i.e. banks) comply with lending standards. The lending standards ("leennorm" in Dutch) follow straightforward rules limiting the amount that can be loaned depending on the financial situation of the lender and are the basis for valid loan approval. Regardless of what an AI system indicates, banks must (and do) conform to this lending standard in all cases. The interviewee of the SA indicated that this was the primary method by which the supervisory authority currently ensured a lending consumer was protected. An interesting point was raised that within the lending standards banks might use AI to find cases their traditional systems would not give a credit, but the AI determines as being profitable for the bank. However, this might not always be good for the consumer. Widespread adoption of AI models might thus require reevaluation of the lending standards.

In summary, for consumer credit, banks reported they use AI in conjunction to traditional ("business rules") systems. As a result of the lending standards, what is and isn't allowed for banks by supervisory authorities in terms of offering loans to consumers is currently clearly specified and understandable for both parties. As a result, in terms of explainability the lending standards are the basis (and thereby the explanation) for rejection of most loans of consumers. As for the edge cases where (within the lending standards) newer AI models might give a different recommendation compared to the traditional models of banks, explainable AI would be especially important to give insight into exactly what causes the deviation from traditional models. Due to the current simplicity of the utilized models, this is at the moment not yet a concern, as also stated by the interviewees. Interviewees at a bank indicated that automated explainability towards consumers (loan applicants) is in principle possible due to the high level of interpretability of the models. Currently, in most cases there is a human-in-the-loop (the advisor) who provides the customer with information and acts as a potential ethical safeguard.

4.2 Credit risk management

The AI system of the first bank in the credit risk management use case follows an AIRB (advanced internal rating-based) model for the bank's residential mortgage portfolio (a capital model). It predicts a probability of default for each mortgage customer and a prediction of loss-given-default for each customer. The model uses around 10-15

variables and is based on logistic regression. There is no interaction with any consumer based on the model, it is only used internally. The main stakeholders for explanations are the internal “first line” and the supervisory authority. More advanced AI is expected to potentially be able to lead to better performance, however, the interviewees reported apprehension to use more complex models due to the expected long and time-consuming process to get approval both internally and externally from supervisory authorities.

From the interview with the first SA, it became apparent that regulations such as capital requirements regulations (CRR [28]) heavily determine the boundaries for what type of AI systems can be used in this use case. Predominantly, logistic regression models are used across all financial institutions. Models that are more complex may not meet requirements like traceability and replicability. Another requirement for credit risk models is to demonstrate "experience" in applying a model. In practice, this means that the model must be used as a shadow model for at least three years before approval can be given. Banks are conducting plentiful research and pilots into AI in credit risk, but the regulations are a limiting factor for further implementation. Currently, AI in credit risk does not appear to lead to sufficient benefit compared to the challenge of getting its use approved within the current regulatory framework to make it worthwhile. It was indicated that the bank first to implement a new AI method must assume it takes at least a year and a half before approval is granted.

In summation, in credit risk management strict requirements are heavily embedded in regulations like CRR. Credit risk management forces ‘transparent by design’ models, therefore, xAI is less of an issue as AI models that are not inherently transparent are simply not used. Regulations/supervisory authorities are slow to change on credit risk, possibly to the more international nature and societal importance of regulation in this use case. Changing these regulations to allow for AI systems that are more complex will be an incremental process that takes time and trust in the safety of such systems.

4.3 Anti-money laundering (AML)

For the first bank the use case of anti-money laundering (AML) involved an AI system developed to detect fraudulent activity in corresponding banking transactions. The AI system consists of two algorithms (models): a deduplication algorithm and a classification algorithm. As AML investigators check the flagged transactions, there is a human-in-the-loop. The AML investigator receives explanations (e.g., the most important features leading to a flagging) as part of the outcome of the AI system. The xAI tool SHAP [21] was used with output provided to the investigator. As such, the investigator can be said to be main stakeholder for explanation in this use case. Explanation, in a broader sense, to other stakeholders is done via technical documentation and various internal processes.

The use case of the second bank concerns machine learning (ML) used for transaction monitoring. In the past, transaction monitoring was only done rule-based. Currently, multiple ML models are used in conjunction with a rule-based methodology. For instance, there is a supervised AI model that is used as noise reduction (i.e. reduces false positives) on the output of the rule-based system. Furthermore, there is also a supervised model that gives customers scores based on suspicion of money laundering

practices and an unsupervised anomaly detection AI model. The output of the models is intended for transaction monitoring analysts who have expertise in recognizing integrity risks. These analysts are generally not concerned with assessing the quality of model output, which is done by quality assurance analysts. The ML model output includes extensive information (which can be considered explanation) about suspicious situations, e.g., indicating the most relevant features, as opposed to rule-based systems. This explainability aspect of these (modern ML) models is thus an important part of the subsequent analysis done by the analyst. This analyst also uses a multitude of other data (sources) outside the detection models for further verification. The analyst can be seen as the human-in-the-loop in this use case, and as the most important stakeholder in need of explanation. Notably, results of the ML-models are improved over the traditional models: both fewer false positives and fewer false negatives (thus more suspicious transactions are reported).

Interviewees indicated that both internally for banks, but also for supervisory authorities, a change of mindset is required to transition from the traditional way of thinking in thresholds (contained in business rules), to more probabilistic thinking about the features of an AML case (contained in modern ML methods). With the latter, explanations can be more complex, but should not be of less quality.

The first SA, in the case of AML, is tasked with ensuring that banks comply with the Anti-Money Laundering and Anti-Terrorist Financing Act [29]. Currently, this SA does not impose any requirements on what type of AI system is used for AML as long as it can be properly explained both to the supervisory authority and internally. Exactly what sufficient explanation is for which type of AI system is not defined by the SA but assessed on a case-to-case basis, due to the highly varying contexts in which AI is used. For the time being, there is also no framework in which explainability is defined, which is directly applicable to this use case. In the context of controlled business operations, a bank must be able to explain how its systems work. If a bank cannot explain an AI system, both to the supervisory authority and internally, as there may be uncontrolled business operations the bank does not sufficiently manage its risks.

In summary, AML was indicated to be one of the use cases that can benefit most from AI in terms of improving results while also being the use case in which the supervisory authorities allow the most room for the use of novel AI methods. So far, the issue of explainability did not hinder the deployment of more complex AI systems in this use case. The internal AML analyst/investigator is viewed as the most important stakeholder regarding explanations by the banks. This investigator is trained to work with and understand model output, which can be seen as a form of, or bringing about of, explainability.

4.4 General

One of the main findings, reported throughout the interviews, is that explainable AI is high on the agenda of banks and supervisory authorities. Within banks, it either is or is planned to be an aspect of an ethical framework used within the organization. Such a framework generally builds on existing principles or procedures (not related to AI specifically), but there is a trend towards more unification of principles and a more explicit

focus on AI. For supervisory authorities, explainability is not exclusively an ethical concern, as it is also relevant from a prudential and legal perspective (e.g., a prudential or legal framework such as CRR, lending standards, and the GDPR). As such, explainability is relevant to a wide range of supervisory authorities in the financial sector among which the two in this study, but also including, e.g., data protection supervisory authorities.

The use of complex AI systems by banks is increasing although often still limited, mainly still using basic methods such as logistic regression. The use case of AML is a notable exception where more varied and advanced AI models are used. In the plenary session, the following reasons for the slow adoption of AI were mentioned: 1) The time needed to become familiar with and implement complex models and especially xAI methods (such as SHAP and LIME [21,22]), which have emerged only in the last years. Deciding what xAI method to choose, and how to implement it, is a challenging process as xAI is still developing rapidly and in a short period new methods might make a current xAI method obsolete. 2) Uncertainty as to whether financial regulations (such as lending standards, CRR) or the supervisory authority would allow the use of novel AI. 3) Traditional models are deemed adequate for many use cases. 4) Internal hesitation to implement complex AI systems in customer facing applications. 5) AI systems that are more complex are difficult to maintain and monitor over time.

As for the types of information that can serve as the basis for explanations it could be noted that across all use cases the supervisory authorities indicated they are interested in the full range of types of information, while the interviewees from banks generally indicated only a subset per use case was relevant (see Table 1).

Table 1.: Responses of SAs and banks on the importance of the types of information that can potentially underpin an explanation for supervisory authorities per use case. A plus-sign (+) indicates a positive, a minus-sign (-) a negative, and both (+/-) indicates a partial importance. Note that each of the three banks only partook in two use case interviews, and thus two banks responded per use case, except for the credit risk use case where only interviewees of one bank filled in this list.

	Consumer Credit			**Credit Risk**		**AML**		
	SAs	**Bank**	**Bank**	**SAs**	**Bank**	**SAs**	**Bank**	**Bank**
The reasons, details, or underlying causes of a particular outcome	+	-	+	+	-	+	-	+
The data and features used as input to determine a particular outcome	+	+	+	+	+	+	-	+
The data used to train and test the AI system	+	+	+	+	+	+	-	+
The performance and accuracy of the AI system	+	-	+	+	-	+	+	+
The principles, rules, and guidelines used to design and develop the AI system	+	+	+	+	+	+	+	+
The process that was used to design, develop, and test the AI system	+	+	+	+	+	+	+	+
The process of how feedback is processed	+	-	+	+	-	+	+	+
The process of how explainers are trained	+	+	+	+	+	+	-	+
The persons involved in design, development, and implementation of AI system	+	-	+	+	-	+	-	+/-
The persons accountable for development and use of the AI system	+	+	+	+	+	+	-	+

5 Discussion and conclusions

The main finding of this study is that there appears to be a disparity between the supervisory authorities (SAs) and the banks regarding the desired scope of explainability required for the use of AI in finance. This is exemplified by responses by these two types of organization on what types of information are required by SAs in the various use cases (visible in Table 1). SAs indicate all types of information are relevant while

banks indicate only a subset is relevant. Various laws and regulations already explicitly or implicitly impose requirements on the explainability of information systems, regardless of whether they are AI systems or other classes of systems. However, the use of AI systems brings with it a new type of ethical, social, and legal challenges in addition to the direct technical challenge of opening the black box of non-interpretable models [8,9,23,30]. Therefore, it seems warranted to further explore how this disparity should be addressed.

The financial sector could perhaps benefit from clear differentiation between technical (model) explainability requirements and explainability requirements of business operations, applicable laws, and regulations on this topic of AI. A similar bifurcation as can be made for transparency (process transparency and model transparency [23]) might be useful for the xAI field: for instance, "AI model explainability" and "AI system explainability". The first of these relating to a set of techniques and methods that are directly used to better understand the AI model and how its input relates to its output. The second of these relating to the broader concept of explainability that views the AI model as embedded in a system or a set of systems or processes. Whether a black box houses a deterministic machine learning system, or whether a (larger) black box houses a complex system of processes and various agents interacting with an AI, both require explanation [25]. In the first case the questions will be more like "how does this input lead to this output", the opening of the traditional black box AI. However in the second case questions could be: "how is this process designed?" or "who is responsible for the data quality?".

Most interviewees, especially the technical (i.e. model developers) associated explainable AI with the technical tools that have been developed in the last few years, that focus on explaining the model in a low-level fashion. While technical tools, such as SHAP [21], give additional information about the operation of a model, they do not answer how such information in general is conveyed understandably to a stakeholder, by means of an explanation suited to that stakeholder [9]. Additionally, these tools are often post-hoc or after the fact [13]. Like requirements as privacy, security, and fairness, explainability should require attention from the onset of the design of an information system, "explainability by design" [31,32].

It should be noted that several factors could have made the disparity (seen in Table 1) larger than it is in actuality. Firstly, the interviewees at the bank might not have the same understanding about the laws and regulations as interviewees from the SAs had. Another explanation for the disparity is that it is difficult to translate laws and regulations into precise requirements for information systems and AI systems in particular [33], thus for novel developments very broad ranges of requirements are assumed. The exact reason for the disparity found in this study is certainly a worthwhile topic of future research as well as for subsequent coordination and collaboration between supervisory authorities and regulated entities on topics such as transparency, explainability, and associated definitions.

The requirements regarding explainable AI reported in the interviews varied widely per use case and stakeholder. This limits the possibility of quickly creating a generic framework or checklist for AI in finance that covers all or most bases. Subsequent research could first explore a single use case to create a full picture of the explanation

requirements and what information is relevant for which stakeholder given a range of possible AI models. Subsequently, mapping stakeholders to xAI methods [19,21,22] to see how they can be helped can be a valuable avenue of research that can produce practical instruments for the implementation of xAI.

This study has several limitations that should be noted. First, we only interviewed employees of a subset of the Dutch financial sector, three banks and two supervisory authorities. In addition, we only spoke to a total of 21 employees across the five organizations. Furthermore, we only touched the surface in the examination of the use cases with interviews as the main method to collect data. More in-depth studies are necessary to confirm and extend our findings and to determine whether our findings hold across different geographies.

We found banks are hesitant to put complex AI models into practice in their primary business processes for the lending and credit risk use cases. Interestingly, supervisory authorities indicated that they in principle do not restrict the use of specific types of AI systems. However, laws and regulations such as lending standards and CRR impose explainability requirements which limit the choice of AI methods beforehand. This might be a chicken-or-the-egg type problem, in which banks are unclear what regulators would precisely allow and therefore do not develop a certain AI solution (based on a certain model), while regulators wait for banks to put AI systems into practice before they can clearly say which type of model is allowed and which is not. To counteract this, in the plenary session it was proposed to increase communication between banks and SAs, also in the development process of new AI models.

Notably, in the consumer credit and AML use cases, the use of novel AI methods went hand in hand with the ability to leverage more (types) of data in addition to the ability to use historical data. This is a clear advantage of these novel AI methods over the traditional business rule systems and might explain the increased performance that was reported in these use cases.

The application of AI at banks for the three use cases is currently only focused on internal stakeholders, such as the investigators in the AML use case or the mid-office employees in the consumer credit use case. The fact that there is a human-in-the-loop was reported as a positive, as this offered an additional safeguard before action was taken based on the AI output. In the future, more familiarity with (fully) automated AI systems might lead to banks deploy more customer-oriented AI.

This is one of the first studies that provides practical insight in the application of xAI in the context of use cases and AI systems in the financial sector. It demonstrates that a wide range of aspects requires attention when designing and building AI systems, and that explainability cannot be considered as a merely technical challenge nor a one-size-fits-all solution. For financial law and policy makers, this research illustrates that financial laws and regulations have an impact on the design of information systems and in particular, AI systems.

In conclusion, there appears to be a disparity between the perspectives as provided by the interviewees of the banks and those of the supervisory authorities for the use cases investigated in this study. Namely, the supervisory authorities view explainability of AI in a wider fashion. Potentially, this can be reframed as the supervisory authorities requiring explanation of the AI model as embedded in a broader system, explicitly or

implicitly part of financial laws and regulations. On the other hand, the regulated entities (i.e. the banks in this study) tended to view explainable AI more as a requirement of only the AI model. A clear differentiation between technical AI (model) explainability requirements and explainability requirements of the wider AI system in relation to applicable laws and regulations can potentially be of benefit to the financial sector and help in the communication between supervisory authorities and banks.

References

1. Schwab, K.: The fourth industrial revolution, Random House LCC US (2017).
2. Zhang, D., Mishra, S., Brynjolfsson, E., Etchemendy, J., Ganguli, D., Grosz, B., Lyons, T., Manyika, J., Niebles, J.C., Sellitto, M., Shoham, Y., Clark, J., Perrault, R.: The AI Index 2021 Annual Report. arXiv preprint arXiv:2103.06312 (2021).
3. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, vol. 58, pp. 82-115 (2020).
4. Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B.: Definitions, methods, and applications in interpretable machine learning. In: Proceedings of the National Academy of Sciences, vol. 116, no. 44, pp. 22071-22080 (2019).
5. European Commission: Proposal for a regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts", https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206, last accessed 2021/06/12.
6. Van der Cruijsen, C., De Haan, J., Roerink, R.: Financial knowledge and trust in financial institutions, Netherlands Central Bank, Research Department (2019).
7. Giudici, P., Hochreiter, R., Osterrieder, J., Papenbrock, J., & Schwendner, P. (2019). AI and financial technology. Frontiers in Artificial Intelligence, 2, 25.
8. Bauer, K., Hinz, O., Van der Aalst, W., Weinhardt, C.: Expl(AI)n It to Me–Explainable AI and Information Systems Research. Business & Information Systems Engineering, vol. 63, no. 2 (2021).
9. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, vol. 267, pp. 1–38 (2019).
10. Mueller H., Ostmann, F.: AI transparency in financial services, The Alan Turing Institute, https://www.turing.ac.uk/news/ai-transparency-financial-services, last accessed 2021/05/28.
11. The High-Level Expert Group on Artificial Intelligence, Ethics Guidelines for Trustworthy A, EU Document, https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai, last accessed 2021/05/21.
12. Belle, V., & Papantonis, I. (2021). Principles and Practice of Explainable Machine Learning. *Frontiers in Big Data*, *4*, 688969. https://doi.org/10.3389/fdata.2021.688969
13. Adadi A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence. IEEE Access, vol. 6, pp. 52138-52160 (2018).
14. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM computing surveys (CSUR), vol. 51, no. 5, pp. 1-42 (2018).

15. Van den Berg, M., Kuiper, O.X.: XAI in the Financial Sector, https://www.internationalhu.com/research/projects/explainable-ai-in-the-financial-sector, last accessed 2021/04/08.
16. McWaters, R., Blake, M., Galaski, R.: Navigating Uncharted Waters: A Roadmap to Responsible Innovation with AI in Financial Services. Part of the Future of Financial Services Series. World Economic Forum (2019).
17. ICO (Information Commissioner's Office) and Alan Turing Institute, Explaining decisions made with AI, https://ico.org.uk/for-organisations/guide-to-data-protection/key-data-protection-themes/explaining-decisions-made-with-ai/, last accessed 2021/04/14.
18. Xie, N., Ras, G., Van Gerven, M., Doran, D.: Explainable deep learning: A field guide for the uninitiated. arXiv Preprint arXiv:2004.14545 (2020).
19. Lipton, Z.C.: The Mythos of Model Interpretability: in machine learning, the concept of interpretability is both important and slippery. Queue, vol. 16, no. 3, pp. 31-57 (2018).
20. Schwalbe, G., & Finzel, B. (2021). XAI Method Properties: A (Meta-)study. *ArXiv:2105.07190 [Cs]*. http://arxiv.org/abs/2105.07190
21. Lundberg, S., Lee, S.I.: A unified approach to interpreting model predictions. arXiv preprint arXiv:1705.07874 (2017).
22. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.
23. Gerlings, J., Shollo, A., Constantiou, I.: Reviewing the Need for Explainable Artificial Intelligence (xAI), in Proceedings of the 54th Hawaii International Conference on System Sciences, pp. 1284-1293 (2021).
24. Confalonieri, R., Coba, L., Wagner, B., Besold, T.R.: A historical perspective of explainable Artificial Intelligence. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 11, no. 1, e1391 (2021).
25. Xu, W.: Toward human-centered AI: a perspective from human-computer interaction. Interactions, vol. 26, no. 4, pp. 42-46 (2019).
26. Mueller, S.T., Hoffman, R.R., Clancey, W., Emrey, A., Klein, G.: Explanation in human-AI systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable AI. arXiv preprint arXiv:1902.01876 (2019).
27. Zhang, B.Z., Ashta, A., Barton, M.E.,: Do FinTech and financial incumbents have different experiences and perspectives on the adoption of artificial intelligence? Strategic Change, vol. 30, no. 3, pp. 223-234 (2021).
28. Joosen, B. P. (2015). Regulatory capital requirements and bail in mechanisms. In Research handbook on crisis management in the banking sector. Edward Elgar Publishing.
29. Anti-Money Laundering and Anti-Terrorist Financing Act (*Wet ter voorkoming van witwassen en financieren van terrorisme*) https://wetten.overheid.nl/BWBR0024282/2021-07-01, last accessed 10-09-2021
30. Dwivedi, Y.K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., Duan, Y., Dwivedi, R., Edwards, J., Eirug, A., Galanos, V., Vigneswara Ilavasaran, P., Janssen, M., Jones, P., Kumar Kar, A., Kizgin, H., Kronemann, B., Lal, B., Williams, M.D.: Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. International Journal of Information Management, 101994 (2019).
31. Leijnen, S., Aldewereld, H., Van Belkom, R., Bijvank, R., Ossewaarde, R.: An agile framework for trustworthy AI. In: NeHuAI@ ECAI, pp. 75-78 (2020).

32. Köhl, M.A., Baum, K., Langer, M., Oster, D., Speith, T., Bohlender, B.: Explainability as a non-functional requirement. In: 2019 IEEE 27th International Requirements Engineering Conference, pp. 363-368 (2019).
33. Siena, A., Mylopoulos, M., Perini, A., Susi A.: From laws to requirements. In: 2008 Requirements Engineering and Law, pp. 6-10 (2008).
34. Van der Burgt, J.: General Principles for the use of AI in the Financial Sector, https://www.dnb.nl/actueel/algemeen-nieuws/dnbulletin-2019/dnb-komt-met-richtlijnen-voor-gebruik-kunstmatige-intelligentie/, last accessed 2021/05/21.
35. Buckley, R.P., Zetzsche, D.A., Arner, D.W., Tang, B.W.: Regulating artificial intelligence in finance: Putting the human in the loop. The Sydney Law Review, vol. 43, no. 1, pp. 43–81 (2021).
36. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, vol. 1, no. 5, pp. 206-215 (2019).

Dutch SQuAD and Ensemble Learning for Question Answering from Labour Agreements

Niels J. Rouws[1,2], Svitlana Vakulenko[1], and Sophia Katrenko[2]

[1] University of Amsterdam, Amsterdam 1098 XG, The Netherlands
[2] DEUS B.V., Amsterdam 1017 DL, The Netherlands
{niels.rouws, sophia.katrenko}@deus.ai
https://deus.ai

Abstract. The Dutch Ministry of Social Affairs and Employment has to regularly explore the content of labour agreements. Studies on topics such as diversity and work flexibility are conducted on the regular basis by means of specialised questionnaires. We show that a relatively small domain-specific dataset allows to train the state-of-the-art extractive question answering (QA) system to answer these questions automatically. This paper introduces the new dataset, Dutch SQuAD, obtained by machine translating the original SQuAD v2.0 dataset from English to Dutch (made publicly available on https://gitlab.com/niels.rouws/dutch-squad-v2.0). Our results demonstrate that it allows us to improve domain adaptation for QA models by pre-training these models first on this general domain machine-translated dataset. In our experiments, we compare fine-tuning the pre-trained Dutch versus multilingual language models: BERTje, RobBERT, and mBERT. Our results demonstrate that domain adaptation of the QA models that were first trained on a general-domain machine-translated QA dataset to the Dutch labour agreement dataset outperforms the models that were directly fine-tuned on the in-domain documents. We also compare several ensemble learning techniques and show how they allow to achieve additional performance gain on this task. A new approach of string-based voting is introduced and we showed that it performs on par with a previously proposed approach.

Keywords: extractive question answering · domain adaptation · Dutch.

1 Introduction

The state of the art in natural language processing (NLP) field has progressed since the introduction of Transformer-based models [25]. BERT [7], one of these models, has become a baseline on numerous benchmarks due to its performance on them [22]. While language models pre-trained on English corpora are common, other languages have fewer available resources. Devlin et al. [7] have trained a multilingual BERT model (mBERT) on 104 languages and monolingual BERT models for non-English languages are being investigated, like BERTje [5] for Dutch, for example. The main advantage of these pre-trained language models

is that they can be applied to multiple downstream tasks, including question answering (QA).

The department *Cao Onderzoek en Beleidsinformatie* (COB) of the Dutch Ministry of Social Affairs and Employment regularly investigates the contents of labour agreements to evaluate existing policies or devise new ones [23]. About 30 research studies are conducted each year by the COB, and every study may include up to 80 questions to be answered for each unique labour agreement. Part of these investigations is extracting answers based on the contents of these labour agreements, which can be automated using a QA system. For this purpose, a small curated dataset composed of roughly 250 training examples is created, adopting the same format as SQuAD [20], with questions relevant to these investigations and paragraphs extracted from roughly 100 labour agreements containing the answers. The relevant paragraphs have been collected by running a baseline model, a BERT model trained on SQuAD data, on textual segments of labour agreements that have previously been identified as relevant by domain experts. The final labour agreement dataset is composed of questions regarding topics like diversity or work flexibility and relevant paragraphs from each labour agreement to make up training examples. As labour agreements are legal documents, the language used and overall document structure differ from Wikipedia texts, which are often used as corpora for pre-training language models and is also used to create the SQuAD dataset [20].

Similar datasets are created for the biomedical field, the COVID-QA dataset [17] or BioASQ [24], for example, where Jeong et al. [10] or Poerner et al. [19] apply transfer learning methods to increase performance on these datasets. Another instance where transfer learning is applied is by Hazen et al. [8] that train general domain QA models to an auto manual dataset with limited data.

This paper will compare the performance of three pre-trained language models on extractive QA for Dutch labour agreements. Three models will be considered: BERTje [5], RobBERT [6], and multilingual BERT (mBERT) [7]. These models will be trained and compared on a general domain using a SQuAD v2.0 dataset [21] which is machine translated into Dutch. The quality of the dataset will be investigated, as well as the impact of further processing on overall performance.

Fine-tuning the trained models to the domain-specific labour agreement (CAO) dataset and ensemble models will be other points of investigation. Models are expected to benefit from training on a large general domain first before being fine-tuned on the labour agreement dataset, a small domain-specific dataset. Ensembles are expected to further improve performance. Furthermore, constructing ensembles with models that excel in different types of queries will perform better than ensembles made up of identical model types [2].

The main research question addressed in this paper is:

– How do pre-trained language models perform on extractive question answering for Dutch labour agreement by using fine-tuning?

Fine-tuning the model from a machine translated general domain Dutch SQuAD v2.0 to a specific domain makes it relevant to answer the sub-questions:

Table 1. Example question-answer pairs from SQuAD v2.0 [21] and the Dutch labour agreement dataset. Question 1 is an answerable, or positive, example, answered by the span of text in red. Question 2 on the other hand is an unanswerable, or negative, example without a valid answer present in the reference text. Question 3 originates from the labour agreement dataset, where the reference text commonly contains elements structuring documents.

Article	Normans
Reference text	The English name "Normans" comes from the French words Normans/Normanz, plural of Normant, modern French normand, which is itself borrowed from Old Low Franconian Nortmann "Northman" or directly from Old Norse Norðmaðr, Latinized variously as Nortmannus, Normannus, or Nordmannus (recorded in Medieval Latin, 9th century) to mean "Norseman, Viking".
Question 1	When was the Latin version of the word Norman first recorded?
Answer	9th century
Question 2	When was the French version of the word Norman first recorded?
Answer	*No answer*

Article	Labour agreement
Reference text	3.2 Arbeidsduur 3.2.1 Basisarbeidsduur De basisarbeidsduur is gemiddeld 36 uur per week en 1872 uur per jaar. 3.2.2 Andere arbeidsduur Je kunt met je leidinggevende een andere arbeidsduur afspreken. De maximale arbeidsduur is gemiddeld 40 uur per week en 2080 uur per jaar. Je loopbaan mogelijkheden worden niet belemmerd door een kortere arbeidsduur
Question 3	Wat is de referteperiode?
Answer	per jaar

- What is the influence of language filtering on a machine translated Dutch SQuAD v2.0?
- How does domain adapting QA models, trained on a general domain dataset to a specific domain, using fine-tuning compare to directly fine-tuning models on a specific domain?

Furthermore, the effectiveness of ensemble models in other applications raises the question:

- What will be the influence of ensemble models on the performance of extractive QA on Dutch labour agreements?

The contributions of this work include the evaluation of Dutch QA models trained on both a general domain and small specific domain. A fine-tuning strategy is employed which can act as an example for other Dutch QA applications

with specific target domains using only a limited amount of data. Furthermore, an analysis and proposed filtering for a machine translated Dutch SQuAD v2.0 dataset is performed. The machine translated Dutch SQuAD v2.0 with additional language filtering is made publicly available[3]. This dataset can still be improved upon to reduce noisy examples due to translation in order to create better Dutch datasets for future studies on extractive QA and other downstream tasks. Finally, Dutch pre-trained language models are compared on the downstream extractive QA task on this Dutch dataset both individually and ensemble learning for small gains in exchange for more computational power.

2 Related work

Training and evaluating Dutch QA systems with a lack of dedicated resources has been investigated by Isotalo [9]. Experiments show that machine translating datasets is a viable option to train Dutch QA systems on. Disadvantages of using machine translated data include reducing linguistic richness of translated texts, possibly resulting in easier examples. Similar works exist that study transfer learning, or domain adaption, of BERT-based models to specific domains like the COVID-QA dataset [17, 19], biomedical QA [10], or QA on an automobile manual domain [8].

Möller et al. [17] has created a QA dataset with 2k examples related to COVID-19 annotated by experts of biomedical sciences. Answers are generally longer and need to be extracted from longer reference texts compared to the general domain SQuAD dataset [20]. A RoBERTa model [13] was fine-tuned on SQuAD and evaluated on the COVID-QA dataset as baseline. EM and F1 scores were both significantly improved on by training the fine-tuned model on the COVID-QA dataset [17]. Poerner et al. [19] propose a CPU-only domain adaptation method for pre-trained language models. This approach involves learning Word2Vec [16] embeddings for text of the target domain, aligning them with the already existing embeddings of the pre-trained language model and updating the embedding layer together with a new tokenizer. A baseline BERT model trained of the SQuAD dataset [20] was adapted using this approach and performs better than prior being domain adapted.

Another example of domain adaptation of BERT models to the biomedical field is the work of Jeong et al. [10]. They apply sequential transfer learning to improve performance of models on biomedical QA. Jeong et al. [10] state that fine-tuning models on both the SQuAD dataset [20] and BioASQ [24], a biomedical QA dataset, produces better results than only training on the BioASQ dataset. Furthermore, they show that fine-tuning BioBERT on natural language inference (NLI), using the MultiNLI dataset [26], followed by training on BioASQ outperforms the SQuAD approach. Additional experiments show that the order of datasets used to fine-tune matters for longer chains fine-tuning on both the MultiNLI and SQuAD datasets prior to BioASQ.

[3] https://gitlab.com/niels.rouws/dutch-squad-v2.0

Hazen et al. [8] investigate domain adaptation to apply QA in new specific domains like an automobile manual. Their standard approach to transfer a QA model to this domain is to use general domain datasets like SQuAD [20] as starting points and training for 2 epochs to the auto manual domain. With limited data, around 200 examples were shown great performance increase on the specific domain and shows that models trained on large amounts of general data can be transfer learned with limited data of a specific domain [8].

This work will be using a machine translated Dutch SQuAD v2.0 dataset as a general domain dataset in order to adapt the domain to a legal domain using limited data extracted from labour agreements. Machine translating existing English datasets into other languages is a strategy employed by others, for instance, translating SQuAD to Spanish [4], Korean [12], or Persian [1].

3 Datasets

The models are fine-tuned and compared on two Dutch QA datasets. A large general domain machine translated Dutch SQuAD dataset and a small domain specific curated dataset composed of Dutch labour agreements.

3.1 Dutch SQuAD v2.0

Dutch SQuAD v2.0 is a machine translated, using the Google Translate API, version of the original SQuAD v2.0 [21] by Borzymowski [3][4]. Direct translations of the answers were used to find the start tokens in the translated reference text. Question-answer pairs were lost in translation if the translated answer is not present in the translated context [3]. Due to this, around 31 thousand question-answer pairs were removed in the training set of the translated version.

Despite these processing steps, noisy examples remain in the dataset containing foreign words, for example, see Table 2. Examples are either partially translated or contain large pieces of non-Dutch languages.

In order to further reduce noise in the translated dataset, language identification [15] is employed to remove noisy non-Dutch examples using Pythons langid module[5]. An example was removed if either the question or the reference text was classified as non-Dutch.

[4] https://github.com/borhenryk/train_custom_qa_model
[5] https://github.com/saffsd/langid.py

Table 2. Examples of contexts in the Dutch SQuAD v2.0 dataset removed using language identification.

Example 1:	After the Peace of Westphalia, several border territories were assigned to the United Provinces. They were federally-governed Generality Lands (*Generaliteitslanden*). They were *Staats-Brabant* (present North Brabant), *Staats-Vlaanderen* (present Zeeuws-Vlaanderen), *Staats-Limburg* (around Maastricht) and *Staats-Oppergelre* (around *Venlo*, after 1715).
Example 2:	New Delhi *is de thuisbasis van* Indira Gandhi Memorial Museum, National Gallery of Modern Art, National Museum of Natural History, National Rail Museum, National Handicrafts and Handlooms Museum, National Philatelic Museum, Nehru Planetarium, Shankar's International Dolls Museum. *en* Supreme Court of India Museum.

Figure 1 shows the language distributions of both questions and answers in the Dutch SQuAD v2.0 training set. Answers are predominantly classified as English followed by Dutch and German, unlike the reference texts and questions that are predominantly Dutch. Out of 18.6k contexts, only 31 cases were classified as non-Dutch in the training set and 3 in the development set, two cases are shown in Table 2.

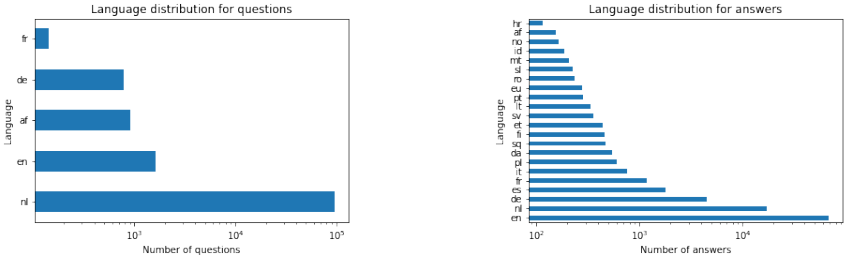


Fig. 1. Language distribution for questions and answers of the Dutch SQuAD v2.0 dataset. All languages are shown that exceed the threshold value $t = 100$.

The exact distribution of example types per dataset are shown in Table 3. Positive examples decrease each iteration, while the number of articles remains constant. The amount of negative examples only decline at the last iteration when they belong to non-Dutch questions or contexts. As a result of both translating and filtering, the proportion of positive to negative examples has shifted towards more negatives per positive example compared to the original SQuAD v2.0.

Table 3. Dataset statistics of SQuAD v2.0 [21], a Dutch SQuAD v2.0, a Dutch SQuAD v2.0 with additional language filtering (LF), and labour agreement dataset.

	English SQuAD v2.0	Dutch SQuAD v2.0	Dutch SQuAD v2.0 (LF)	Labour agreements
Train				
Total examples	130,319	99,265	95,054	241
Positive examples	86,821	55,767	53,376	165
Negative examples	43,498	43,498	41,768	76
Development				
Total examples	11,873	9,669	9,294	103
Positive examples	5,928	3,724	3,588	71
Negative examples	5,945	5,945	5,706	32

3.2 Labour agreement dataset

The labour agreement (CAO) dataset is a domain-specific dataset with almost 250 training examples collected from close to 100 labour agreements of Dutch businesses. Question-answer pairs were collected and curated in cooperation with experts from the Dutch Ministry of Social Affairs and Employment. Labour agreements are legally binding contracts, which is reflected in the language used in both questions and reference texts, which are relatively short compared to the SQuAD v2.0 dataset [21]. Negative examples are composed of rejected combinations of questions and reference texts. They are added to have slightly more than two positive examples per negative example, as is the case in the training set of SQuAD v2.0.

4 Approach

The different model configurations and training policies will be described that were applied to BERTje [5], RobBERT [6], and mBERT [7] in order to make meaningful comparisons.

4.1 Fine-tuning

Initially, the three models were trained on both the unfiltered Dutch SQuAD v2.0 dataset and the language filtered Dutch SQuAD v2.0 dataset to test whether an mBERT would have an advantage due to translation errors. The fine-tuning strategy for all experiments consist of training the models for 2 epochs with a learning rate of 5e−5, batch size of 8 and AdamW [14] with ϵ = 1e−8. The pre-trained models were acquired from the Hugging Face model database [6][7][8] and used as starting points for baseline models on the CAO dataset, and the

[6] https://huggingface.co/GroNLP/bert-base-dutch-cased
[7] https://huggingface.co/pdelobelle/robbert-v2-dutch-base
[8] https://huggingface.co/bert-base-multilingual-cased

models trained on the SQuAD datasets for testing whether filtering the dataset would improve the results of the monolingual models relative to the multilingual model. The models fine-tuned on the filtered Dutch SQuAD v2.0 dataset are subsequently fine-tuned for another 2 epochs on the CAO dataset and compared to the baselines.

4.2 Voted BERT

In order to boost performance on the CAO dataset, two ensemble approaches utilizing voting mechanisms have been implemented.

The first approach votes based on the sub-strings enclosed by the output answer spans of models. Voting for the second approach, on the other hand, relies on the output scores produced by the dot product of token scores with the start and end vectors. Score voting is applied to ensemble identical models and string voting to combine mixed model types due to the different tokenizers and vocabularies of different models.

Score-based voting A model fine-tuned on the filtered Dutch SQuAD v2.0 dataset is copied K times. Each model k is independently fine-tuned, following the general strategy, on the CAO dataset with a unique seed. At evaluation time, the models are combined into an ensemble that makes prediction based on the output scores of the K models. The output of a single model k is a start vector s_k and an end vector e_k of size l which is the maximum sequence length. s and e contain the logits that denote the probability of tokens in the input sequence being the start and end symbols of an answer. These probabilities are summed and normalized by K to produce the start and end vector representing the prediction of the ensemble [27]. If $\text{BERT}(x;\theta_n)$ denotes the tuple $\langle s_k, e_k \rangle$ predicted by a BERT model with parameters θ_k from the input x, this ensemble can be formulated as:

$$\text{BERT}_{VOTE}(x;\Theta) = \frac{1}{K}\sum_{k=1}^{K}\text{BERT}(x;\theta_k) \tag{1}$$

String-based voting The other voted BERT approach is implemented by voting using an algorithm comparing the output strings in order to mix different BERT models. As for the score-based approach K models are fine-tuned on the CAO dataset, they are, however, different model types. One model for each type is fine-tuned and combined with the others at evaluation time. A naive variant of the algorithm votes for the most occurring exactly matching output, or defaults to the longest available prediction in the voting pool. The other version does not require the outputs of individual models to match exactly. It calculates the longest common sub-string [9] for each unique combination and votes on the longest prediction in the highest scoring combination.

[9] https://www.geeksforgeeks.org/longest-common-substring-dp-29/

5 Evaluation

A description of the pre-trained language models that have been experimented on will be given in this section, in addition to the evaluation metrics used to assess and compare them.

5.1 Models

The pre-trained language models used are comparable in parameters and architecture but vary in, for example, corpora and objectives during pre-training.

BERTje BERTje [5] is a Dutch monolingual model comparable to BERT_{base} with 12 layers and cased tokenization. It has a vocabulary of 30k tokens and is pre-trained on 12 GB of corpora originating from Dutch books, TwNC, SoNaR-500, Web news, and Wikipedia. It is pre-trained on two objectives: sentence order prediction (SOP) and masked language modelling (MLM). For their MLM objective, they mask consecutive word pieces that belong to the same word instead of randomly masking single word pieces.

RobBERT Another monolingual model is RobBERT [6], a Dutch RoBERTa based model with 12 self-attention layers, 12 heads and 117M parameters. RobBERT is pre-trained using the RoBERTa training regime [13] and does not include the SOP objective compared to BERTje. The OSCAR corpus was used as a dataset, which is 39 GB of Dutch text obtained from the Common Crawl corpus. It also includes their own byte pair encode (BPE) tokenizer constructed using the OSCAR corpus consisting of 40k tokens, 10k more than BERTje. The authors found that RobBERT outperforms other BERT-like models when dealing with smaller datasets.

mBERT mBERT [7] is a multilingual model for 104 languages trained using Wikipedia texts using an MLM objective and next sentence prediction (NSP). mBERT can generalize across languages with a multilingual representation of words without an explicit training objective for this task [18].

5.2 Evaluation metrics

We evaluated the QA models using two metrics: exact match (EM) and F1 scores. In addition to calculating EM and F1 scores on the complete datasets, scores are calculated for both the subsets of data containing only positive examples (HasAns) and negative examples (NoAns) individually to give a better insight into the performance of the models.

Moreover, we calculated the EM and F1 scores per interrogative Dutch words to gain an understanding of challenging questions. Models that excel at different

question types can be combined in an ensemble to exploit strengths and compensate for weaknesses. Question types were assigned to questions by using regular expressions for Dutch interrogative words: *wie, wat, waar, waarom, wanneer, welk, welke, hoe, hoeveel*. Questions without a match for any of these words were placed in a separate category: *other*.

6 Results

This section presents the collected results to answer the research questions, starting off with the results generated from the Dutch SQuAD v2.0 dataset, followed by the results of QA systems on the labour agreement dataset.

6.1 Dutch SQuAD

Language filtering The effect of language filtering described in section 3.1 is tested by fine-tuning BERTje [5], RobBERT [6], and mBERT [7] models on both the unfiltered and language filtered Dutch SQuAD v2.0 dataset and evaluating these models on their respective development sets. The results of this experiment are shown in Table 4 with models trained on the language filtered dataset followed by (LF). The HasAns column show the scores calculated exclusively on the subset of positive examples and NoAns scores on the subset of negative examples. mBERT achieves the highest scores on the unfiltered dataset by a large margin on all subsets of the data. While remaining the best performing model, the difference between models shrinks as RobBERT's scores improve on all fields and BERTje slightly decreases except on the NoAns section, where it becomes the best scoring model.

Table 4. Evaluation results of models, on their respective development set, fine-tuned on the unfiltered Dutch SQuAD v2.0 dataset and language filtered version. Models fine-tuned on the language filtered version are followed by (LF). The HasAns column are the evaluation scores exclusively with the subset of positive examples and NoAns scores on the subset of negative examples. **Bold** font indicates the best scores on the unfiltered dataset, and underlined font indicates the best scores on the filtered dataset.

Model	EM / F1	HasAns EM / F1	NoAns F1
BERTje	65.26 / 69.13	44.33 / 54.39	78.37
BERTje (LF)	65.05 / 68.72	43.62 / 53.89	78.53
RobBERT	63.38 / 67.34	43.43 / 53.72	75.88
RobBERT (LF)	64.64 / 68.55	45.43 / 55.54	76.73
mBERT	**67.37** / **71.31**	**47.80** / **58.03**	**79.63**
mBERT (LF)	65.69 / 69.35	46.40 / 55.89	77.81

Results per question type The datasets contain a diverse mix of question types, which have been evaluated as separate subsets to identify challenging

questions and compare whether the challenge exists across model types. Table 5 contains these results for all positive examples of the language filtered Dutch SQuAD v2.0 development set for the models fine-tuned on the training set.

All three models show a comparable performance distribution along the question types. *Wie*/Who and *wanneer*/when questions are among the best performing types, while *waarom*/why, *hoe*/how, and other questions score worst and have significantly large differences between EM and F1 scores. Predicting the ground truth for these question types appears to be challenging, but still parts of them are captured relatively frequently. *Wat*/What scores are surprisingly low for the high number of examples compared to other questions.

Table 5. Model scores of positive examples evaluated per question type on the filtered Dutch SQuAD v2.0 development set. Underlined scores denote the highest scores per row, and **bold** scores the highest score for a model type.

Question type	Number of examples	BERTje HasAns EM / F1	mBERT HasAns EM / F1	RobBERT HasAns EM / F1
wie/who	332	**59.34** / **65.41**	61.14 / 67.09	**61.45** / 68.39
wat/what	1035	35.65 / 45.23	38.16 / 47.47	36.23 / 46.56
waar/where	244	35.66 / 49.35	35.66 / 52.51	38.52 / 51.87
waarom/why	44	20.45 / 41.95	22.73 / 35.22	11.36 / 33.61
wanneer/when	289	55.36 / 61.67	**65.74** / **73.51**	61.25 / **69.45**
welk/which	444	50.90 / 57.49	52.70 / 59.56	53.15 / 59.06
welke/which	629	44.67 / 53.10	47.22 / 54.55	46.10 / 55.42
hoe/how	198	33.33 / 47.30	37.88 / 52.00	36.87 / 53.87
hoeveel/how much	324	50.00 / 63.49	50.62 / 63.66	51.85 / 65.45
other	103	25.24 / 37.51	30.10 / 42.58	26.21 / 36.15

6.2 Labour agreement dataset

Domain adaptation Table 6 shows the results of all systems trained on the labour agreement (CAO) dataset. The training strategies can be derived from the datasets following the model name. Baseline models are fine-tuned on the CAO dataset only, as opposed to domain adapted models. They are first fine-tuned on the large general domain language filtered Dutch SQuAD v2.0 (DSQuAD) followed by fine-tuning on the small domain specific CAO dataset. The results show that the baseline models are outclassed by the domain adapted version of the same model type. BERTje mainly gains performance on the negative examples and sees the least improvement on the positive examples, whereas both RobBERT and mBERT drop performance for negative examples and gain significant performance on positive examples. In addition to outperforming baseline models, domain adapted models attain higher scores on the CAO dataset than models score on the Dutch SQuAD datasets (see Table 4).

Ensemble models Ensemble models show in the majority of cases an increase in performance regarding single models. The score-based approach with ensem-

Table 6. Exact match (EM) and F1 scores of all systems evaluated on the CAO development set. **Bold** scores indicate the highest score per column, and underlined scores indicate the highest score per model type. Baseline models are fine-tuned on the CAO dataset (CAO) while all other systems are first fine-tuned on the filtered Dutch SQuAD v2.0 dataset followed by fine-tuning on the CAO dataset (DSQuAD + CAO). K denotes the ensemble size of score-based voted BERT systems. The final cell shows the mixed ensembles using string-based voting with LCS to indicate voting using the longest common sub-string algorithm.

System	EM / F1	HasAns EM / F1	NoAns F1
BERTje (CAO)	62.14 / 65.99	57.75 / 63.34	71.88
BERTje (DSQuAD + CAO)	66.02 / 71.38	59.15 / 66.93	**81.25**
BERTje (DSQuAD + CAO) (K=3)	66.99 / 73.94	60.56 / 70.65	**81.25**
BERTje (DSQuAD + CAO) (K=5)	65.05 / 72.78	61.97 / 73.19	71.88
RobBERT (CAO)	58.25 / 61.15	50.70 / 54.90	75.00
RobBERT (DSQuAD + CAO)	66.99 / 73.48	66.20 / 75.61	68.75
RobBERT (DSQuAD + CAO) (K=3)	69.90 / **76.83**	66.20 / 76.24	78.13
RobBERT (DSQuAD + CAO) (K=5)	65.05 / 72.78	61.97 / 73.19	71.88
mBERT (CAO)	63.11 / 68.23	59.15 / 66.58	78.13
mBERT (DSQuAD + CAO)	69.90 / 76.38	67.61 / **77.00**	75.00
mBERT (DSQuAD + CAO) (K=3)	69.90 / 75.57	66.20 / 74.42	78.13
mBERT (DSQuAD + CAO) (K=5)	**70.87** / 75.90	**69.01** / 76.30	75.00
BERTje + RobBERT + mBERT (DSQuAD + CAO)	**70.87** / 76.28	67.60 / 75.45	78.13
BERTje + RobBERT + mBERT (DSQuAD + CAO) (LCS)	69.90 / 76.47	66.20 / 75.72	78.13

ble sizes of $K = 3$ and $K = 5$ produce primarily better results than single models. Increasing the ensemble sizes also appear to benefit scores on positives examples for both BERTje and mBERT. RobBERT, on the other hand, sees a sudden decrease in performance for $K = 5$. The ensembles composed of mixed models perform generally well, achieving high overall scores. Voting using the string matching approach or largest common sub-string (LCS) approach achieve comparable results, with a trade-off between EM scores and F1 scores for positive examples.

7 Discussion

The most significant findings include the improved performance of domain adapted models compared to baseline models and slight additional gain in performance of ensemble models compared to their single model counterparts. These results were expected based on the results of similar studies of transfer learning models from general domain datasets to specific domains in the biomedical domain [17, 10], for example, or on automobile manuals [8].

The ensemble models slightly improve model results as expected [27] which could be improved upon by creating ensembles of models that do not have as similar performance distributions per question type as have been found for BERTje [5], RobBERT [6], and multilingual BERT [7].

Hyperparameter optimization for BERT during fine-tuning could increase model performance. All models have been fine-tuned using a general strategy

which is likely not optimal for each model type, leading to under- or overperforming models.

Improving models for the labour agreement domain could alternatively take the approach of BioBERT [11] by pre-training on data from the target domain. However, pre-training a model on corpora within a domain requires large amounts of data and computing power. Alternatively, the relatively inexpensive domain adaptation approach of Poerner et al. [19] could be explored.

7.1 Conclusion

In this paper, we examined fine-tuning pre-trained language models for a Dutch-language QA task. The models were evaluated on a general-domain machine-translated Dutch SQuAD as well as on a low-resource target domain of Dutch labour agreements. Our results show that fine-tuning the models on the language-specific QA dataset is beneficial even when such dataset is machine translated from English. This finding has important implications beyond the QA task showing that the model performance can be improved across languages by machine translating English-language resources.

We also note, however, that the domain-adapted models using fine-tuning attain higher scores on the labour agreement dataset than on the Dutch SQuAD v2.0 datasets. The cause of this is likely that a machine translated dataset contains more noise compared to a curated dataset. A limited variety of questions for the labour agreement dataset could be another reason why higher scores are attained. Our results demonstrate that the best performance can be achieved by using a mixed ensemble of mBERT, BERTje and RobBERT using string-based voting, closely followed a mBERT ensemble utilizing a score-based voting system. The best models overall reaching EM scores up to 70.87% and a F1 score of 76.28% on the target domain.

Interestingly, language filtering the machine-translated Dutch SQuAD results in decreased performance for mBERT, while RobBERT gained in performance and BERTje had only slight changes in performance. All of these results are still significantly below comparable QA models for English.

Our results provide important insights on the intricacy of domain adaptation for non-English QA models. We show that it is feasible to train QA models in a low-resource scenario which is prevalent when automating recurrent tasks in the real-world settings, such as the labour agreement investigations by the Dutch Ministry of Social Affairs and Employment.

Bibliography

[1] Negin Abadani, Jamshid Mozafari, Afsaneh Fatemi, Mohammd Ali Nematbakhsh, and Arefeh Kazemi. Parsquad: Machine translated squad dataset for persian question answering. In *2021 7th International Conference on Web Research (ICWR)*, pages 163–168. IEEE, 2021.

[2] Anna Aniol, Marcin Pietron, and Jerzy Duda. Ensemble approach for natural language question answering problem. In *2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW)*, pages 180–183. IEEE, 2019.

[3] H. Borzymowski. henryk/bert-base-multilingual-cased-finetuned-dutch-squad2 · Hugging Face, 2020. URL https://huggingface.co/henryk/bert-base-multilingual-cased-finetuned-dutch-squad2.

[4] Casimiro Pio Carrino, Marta R Costa-jussà, and José AR Fonollosa. Automatic spanish translation of the squad dataset for multilingual question answering. *arXiv preprint arXiv:1912.05200*, 2019.

[5] Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. Bertje: A dutch BERT model. *CoRR*, abs/1912.09582, 2019. URL http://arxiv.org/abs/1912.09582.

[6] Pieter Delobelle, Thomas Winters, and Bettina Berendt. Robbert: a dutch roberta-based language model, 2020.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[8] Timothy J. Hazen, Shehzaad Dhuliawala, and Daniel Boies. Towards domain adaptation from limited data for question answering using deep neural networks, 2019.

[9] Laura Isotalo. Generative question answering in a low-resource setting.

[10] Minbyul Jeong, Mujeen Sung, Gangwoo Kim, Donghyeon Kim, Wonjin Yoon, Jaehyo Yoo, and Jaewoo Kang. Transferability of natural language inference to biomedical question answering. *arXiv preprint arXiv:2007.00217*, 2020.

[11] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4): 1234–1240, 2020.

[12] Kyungjae Lee, Kyoungho Yoon, Sunghyun Park, and Seung-won Hwang. Semi-supervised training data generation for multilingual question answering. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[15] Marco Lui and Timothy Baldwin. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P12-3005.

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[17] Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. Covid-qa: A question answering dataset for covid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, 2020.

[18] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert?, 2019.

[19] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. Inexpensive domain adaptation of pretrained language models: Case studies on biomedical NER and covid-19 QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1482–1490, Online, November 2020. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.134. URL https://www.aclweb.org/anthology/2020.findings-emnlp.134.

[20] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[21] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad, 2018.

[22] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works, 2020.

[23] Startup in Residence Intergov. Geautomatiseerde tekstanalyse cao's | Startup in Residence Intergov, 2020. URL https://intergov.startupinresidence.com/nl/szw/geautomatiseerde-tekstanalyse-cao/brief.

[24] George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):1–28, 2015.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[26] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

[27] Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. Improving bert fine-tuning via self-ensemble and self-distillation. *arXiv preprint arXiv:2002.10345*, 2020.

Encore abstracts

BNAIC/BeneLearn proceedings
November 10–12, 2021
Belval, Esch-sur-Alzette (Luxembourg)

A Real-Time Method for Detecting Temporary Process Variants in Event Log Data

Sudhanshu Chouhan[1][0000−0002−9151−0879], Anna Wilbik[2][0000−0002−1989−0301], and Remco Dijkman[1][0000−0003−4083−0036]

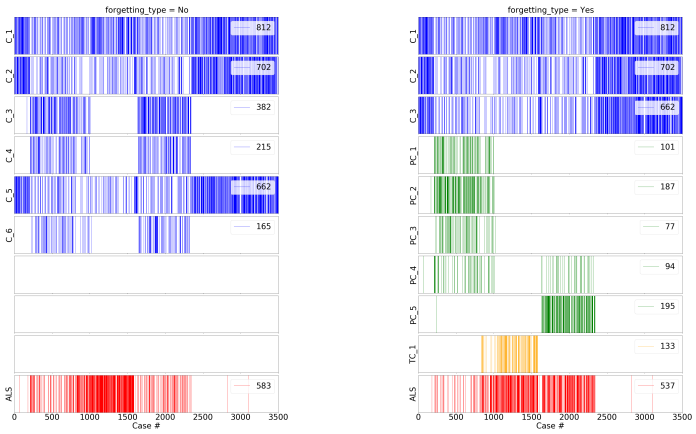
[1] Eindhoven University of Technology, Eindhoven 5612 AZ, The Netherlands
{s.g.r.chouhan,r.m.dijkman}@tue.nl
[2] Maastricht University, Maastricht 6229 GT, The Netherlands
a.wilbik@maastrichtuniversity.nl

During the execution of a business process, organizations or individual employees may introduce mistakes and temporary or permanent changes to the process. Such mistakes and changes in the process can introduce anomalies and deviations in the event logs, which in turn introduce temporary and periodic process variants. While methods exist for detecting anomalous cases in business processes, these methods will not detect different variants of the process. To fill this gap, the method we present in [1] discovers, in real-time, temporary and permanent changes to the process from event log data, in addition to anomalies. The method classifies cases in an event log into four categories: (i) common cases (type of cases which are most-followed in the process), (ii) temporary cases (type of cases which are followed temporarily in the process), (iii) periodic cases (type of cases which are followed at certain times in the process), and (iv) anomalous cases (type of anomalous cases). At the core of this method lies a clustering approach using Non-Euclidean Relational Fuzzy c-Means (NERFCM) [2] supported by Correlation Cluster Validity (CCV) [4].

The proposed method works as follows. First, the user defines the number of initial cases to form initial clusters, the merging criteria, and the number of days after which an existing cluster or anomalies may be forgotten or saved. Then CCV algorithm is applied to estimate the probable number of clusters that exist in initial cases, followed by the application of the NERFCM algorithm, which creates the initial clusters. The clusters are saved in a list called 'cluster list'. At this stage, the cut-off size for a new cluster is also computed, i.e. the size of a new cluster to qualify into the main cluster list. When a new case arrives and falls under the radius of any existing clusters, it is added to that cluster; otherwise, it is stored in a list called 'anomaly list'. Simultaneously, it is checked if new clusters are forming inside the anomaly list. If, at any point in time, the size of a cluster in the anomaly list becomes larger than the defined cut-off size, then that cluster of cases is removed from the anomaly list and added to the main list of clusters. Next, if at any point in time the similarity between any two or more clusters in the cluster list becomes greater than a defined merging criteria, then those clusters are merged. If no new case is added to a cluster in the cluster list for a defined number of days, that cluster is removed from the cluster list and added to a list of forgotten clusters. Similarly, if no new case is added to the anomaly list for the same number of days, all the cases are removed from the

anomaly list and saved as confirmed anomalies. The algorithm then waits for a new case to arrive and implements all the steps again.

We evaluated our method on several synthetic and real-life event logs. To show the effectiveness of the method here we only discuss the results from one of the synthetic event logs where we knew the occurrence of different process variants. Moreover, anomalies were introduced to the event log, using the approach proposed in [3]. Figure 1 shows a visual comparison between results obtained by setting the choice of forgetting the clusters as 'No' and 'Yes'. In Figure 1, each row represents a cluster, where cluster C_n represents common cases, cluster PC_n represents periodic cases, cluster TC_n represents temporary cases, where n is the number of cluster. For instance, C_1 shows the first cluster in the main cluster list. The last row in both Figure 1a and Figure 1b shows the confirmed anomalies (ALS). The horizontal axis shows the arrival of cases in the order of their time of completion. Each vertical bar in a cluster shows the assignment of a case to that cluster. In Figure 1b, PC_1-PC_5 and TC_1 are the clusters that



(a) *forgetting type* = No (b) *forgetting type* = Yes

Fig. 1: Blue bars are common cases, Green bars are periodic case, Orange bars are temporary cases, and Red bars are cases marked anomalous.

were forgotten from the main cluster list at some point in time since no new case was added to them. In the post-analysis of the results, it is found that a cluster similar to PC_1 reappeared again in PC_2, PC_3 and part of PC_4. Also, part of PC_4 reappeared in PC_5. Since the reappearing clusters are similar to each other and they were forgotten after some time, therefore, they are categorized as periodic cases. On the other hand, TC_1 is a cluster that was forgotten after some time, but no similar cluster ever reappeared in main clusters or forgotten clusters. Therefore, cases in TC_1 are categorized as temporary cases. Furthermore, in Figure 1a, all the periodic and temporary cases are included in the main cluster. Cases falling in these clusters make up of periodic and temporary process variants.

References

1. Chouhan, S., Wilbik, A., Dijkman, R.: A real-time method for detecting temporary process variants in event log data. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) Business Process Management. pp. 197–214. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-85469-0_14
2. Hathaway, R.J., Bezdek, J.C.: Nerf c-means: Non-euclidean relational fuzzy clustering. Pattern recognition **27**(3), 429–437 (1994)
3. Nolle, T., Luettgen, S., Seeliger, A., Mühlhäuser, M.: Binet: Multi-perspective business process anomaly classification. Information Systems p. 101458 (2019)
4. Popescu, M., Keller, J.M., Bezdek, J.C., Havens, T.: Correlation cluster validity. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics. pp. 2531–2536. IEEE (2011)

Average Localised Proximity: A new data descriptor with good default one-class classification performance (abstract)

Oliver Urs Lenz[1], Daniel Peralta[1,2], and Chris Cornelis[1]

[1] Department of Applied Mathematics, Computer Science and Statistics, Ghent University {oliver.lenz,chris.cornelis}@ugent.be http://www.cwi.ugent.be
[2] Data Mining and Modelling for Biomedicine group, VIB Center for Inflammation Research, Ghent University daniel.peralta@irc.vib-ugent.be https://www.irc.ugent.be

1 Introduction

The goal of one-class classification, also known as semi-supervised outlier, anomaly or novelty detection, is to distinguish between a *target* class and the *other* class, on the basis of a training set that only contains target class instances. Many one-class classification algorithms, known as *data descriptors*, contain one or more hyperparameters that need to be set by the user. Previous experimental comparisons of data descriptors have used instances from the other class to tune these hyperparameter values [4, 12].

The contribution of our paper [6] is threefold. First, we present our own algorithm, Average Localised Proximity (ALP). Second, we determine optimal default hyperparameter values for a number of data descriptors. And third, we compare the performance of a number of data descriptors experimentally.

2 Average Localised Proximity

ALP builds on a number of existing nearest neighbour data descriptors. The simplest of these is Nearest Neighbour Distance (NND) [5], the distance of a test instance to its kth nearest neighbour in the training set. Because the density of the target class may vary throughout the feature space, Localised Nearest Neighbour Distance (LNND) [9, 13] divides this distance by the distance between the kth nearest neighbour and its own kth nearest neighbour in the training set. Unfortunately, this also increases its sensitivity to random fluctuations in the distribution of the training set. Local Outlier Factor (LOF) [1] is based on a more complex calculation that involves three rounds of aggregation and the substitution of small local distances with larger values.

Like LOF, ALP aggregates localised distance values, but is less complex and does not discard any values. ALP has two hyperparameters, k and l. For each $i \leq k$ and each $j \leq l$, we calculate the ith nearest nearest neighbour distance of the jth neighbour of a test instance. We then take the weighted mean with

linearly decreasing weights of the values corresponding to each i, to obtain the local ith nearest neighbour distance, and divide this by the sum of itself and the ith nearest neighbour distance of the test instance to obtain the ith localised proximity value of the test instance. Finally, we sort these proximity values from large to small and again take a weighted mean with linearly decreasing weights.

3 Experiments and results

Our experimental data consists of 246 one-class classification problems derived from 50 real-life datasets from the UCI machine learning repository. Each problem is created by choosing one class as the target class and combining the remaining classes to form the other class. Each feature in the data is rescaled by dividing by the interquartile range of that feature in the training set.

We first determine the optimal default hyperparameter values of ALP, NND, LNND, LOF, as well as the Support Vector Machine (SVM) data descriptor [14, 11], by identifying the values that obtain the highest weighted mean AUROC on our problem set, giving equal weight to each original dataset (Table 1).

Table 1. Optimal default hyperparameter values of data descriptors, with n the size of the target class and m the number of attributes. Hyperparameters k and l rounded to the nearest integer in the range $[1, n-1]$.

Data descriptor	Hyperparameter	Optimal default value
NND	k	1
LNND	k	$3.4 \log n$
LOF	k	$2.5 \log n$
SVM	ν	0.20
	c	$0.25m$
ALP	k	$5.5 \log n$
	l	$6.0 \log n$

Next, we compare the data descriptors with each other, as well as a number of data descriptors that don't require setting any hyperparameter values: Mahalanobis Distance (MD) [8], Isolation Forest (IF) [7], Extended Isolation Forest (EIF) [3], and the Shrink Autoencoder (SAE) preprocessor combined with centroid distance [2]. The hyperparameter values in this comparison are set using a leave-one-dataset-out scheme. Using a clustered Wilcoxon signed-rank test [10] for each pair of data descriptors, and correcting for multiple testing, we find that ALP performs significantly better than LNND, LOF, and the other data descriptors, except SVM, for which the difference is only weakly significant.

Subsequent analysis shows i.a. that ALP has a particularly strong advantage (in general, and over SVM in particular) with one-class classification problems that admit good solutions.

References

1. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying density-based local outliers. In: SIGMOD 2000: Proceedings of the ACM international conference on Management of data. vol. 29, pp. 93–104. ACM (2000)
2. Cao, V.L., Nicolau, M., McDermott, J.: Learning neural representations for network anomaly detection. IEEE Trans Cybern **49**(8), 3074–3087 (2019)
3. Hariri, S., Carrasco Kind, M., Brunner, R.J.: Extended Isolation Forest. IEEE Trans Knowl Data Eng **33**(4), 1479–1489 (2021)
4. Janssens, J.H.M., Flesch, I., Postma, E.O.: Outlier detection with one-class classifiers from ML and KDD. In: ICMLA 2009: Proceedings of the Eighth International Conference on Machine Learning and Applications. pp. 147–153. IEEE (2009)
5. Knorr, E.M., Ng, R.T.: A unified notion of outliers: Properties and computation. In: KDD-97: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. pp. 219–222. AAAI (1997)
6. Lenz, O.U., Peralta, D., Cornelis, C.: Average Localised Proximity: A new data descriptor with good default one-class classification performance. Pattern Recognition **118**, 107991 (2021)
7. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation Forest. In: ICDM 2008: Proceedings of the Eighth IEEE International Conference on Data Mining. pp. 413–422. IEEE (2008)
8. Mahalanobis, P.C.: On the generalized distance in statistics. Proc Natl Inst Sci India **2**(1), 49–55 (1936)
9. de Ridder, D., Tax, D.M.J., Duin, R.P.W.: An experimental comparison of one-class classification methods. In: ASCI'98: Proceedings of the Fourth Annual Conference of the Advanced School for Computing and Imaging. pp. 213–218. ASCI (1998)
10. Rosner, B., Glynn, R.J., Lee, M.L.T.: The Wilcoxon signed rank test for paired comparisons of clustered data. Biometrics **62**(1), 185–192 (2006)
11. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Tech. Rep. MSR-TR-99-87, Microsoft Research, Redmond, Washington (1999)
12. Swersky, L., Marques, H.O., Sander, J., Campello, R.J.G.B., Zimek, A.: On the evaluation of outlier detection and one-class classification methods. In: DSAA 2016: Proceedings of the 3rd IEEE International Conference on Data Science and Advanced Analytics. pp. 1–10. IEEE (2016)
13. Tax, D.M.J., Duin, R.P.W.: Outlier detection using classifier instability. In: SSPR/SPR 1998: Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition. Lecture Notes in Computer Science, vol. 1451, pp. 593–601. Springer (1998)
14. Tax, D.M.J., Duin, R.P.W.: Data domain description using support vectors. In: ESANN 1999: Proceedings of the Seventh European Symposium on Artificial Neural Networks. pp. 251–256. D-Facto (1999)

Combining Logic and Natural Language Processing to Support Investment Management

Marjolein Deryck[1,2], Nuno Comenda[3], Bart Coppens[3], and Joost Vennekens[1,2]

[1]KU Leuven, Dept. Computer Science, Campus De Nayer
[2]Leuven.AI – KU Leuven Institute for AI, Leuven, Belgium
[3]Coppens and Partners Consulting
{marjolein.deryck,joost.vennekens}@kuleuven.be
{nuno.comenda, bart.jan.coppens}@coppens-and-partners.com

The Knowledge Base Paradigm (KBP) advocates a strict separation between declarative domain knowledge and logical inference tasks that can be applied to this knowledge to solve problems of interest [2]. In this paper we report the results from a case study in which we combine the principles of the KBP with a Natural Language (NL) interface. The case was executed at an international financial institution. As a part of its service, an investment banker offers clients advice on the financial products to buy or sell. The clients' preferences can be expressed in an investment profile, that determines which assets are eligible for a specific investment. The eligibility of a specific asset depends on a plethora of interacting rules and constraints. Previously, a bank operator translated the several requests into lengthy programs that contain a lot of enumerations, repetitions, and complex nesting of if-then clauses and exceptions that need to be followed in the right order. This makes each creation of an investment selection program a complex and time consuming task. Furthermore, the result is hard to validate, which entails a substantial operational risk.

Our application allows the eligibility of financial products to be defined by means of controlled natural language (CNL). Each sentence is constructed from a number of building blocks that are selected step by step to get to a complete sentence. The resulting highly structured NL sentence is automatically translated to first order logic (FOL). The application also contains a deep learning NLP module that accepts free-form English. It proposes three CNL statements that are most likely to present the English sentence. The user then selects the most correct sentence, makes adjustments if necessary, and validates the result.

When completed, the KB can be used by different inference methods to perform multiple tasks in the problem domain. We use the IDP system with its associated FOL-based language as underlying reasoning engine [1]. A KB consists of three parts: a vocabulary that contains the ontology of the domain, a theory that contains rules and constraints on the concepts in the vocabulary, and a structure, that delineates the domain of the concepts, and typically gives an interpretation for some of them. The information that is declaratively stated in the KB, can be used for different purposes. The inference task of *model expansion* can be used to decide on the eligibility of a specific asset. Given a theory T (that contains the rules of eligibility), and an interpretation I_p for part of its vocabulary V, the *model expansion* inference computes interpretations I_t for the entire

V such that $I_p \subset I_t$ and $I_t \models T$ [4]. In our application we typically possess all the information on the asset, such that only the values of *Eligible* and *NotEligible* need to be computed. The *optimize* inference is used to find a combination of eligible assets that can be acquired at minimal cost. To this end we create an additional term m that represents this cost. Given a theory T, interpretation I_p and term m, the *optimize* inference will look for a model expansion I_t of I_p that minimizes m [1]. This is, it will select a combination of assets with the lowest associated cost that follows the eligibility rules and given I_p. The *propagation* inference computes a set of facts that are consequences of T given I_p, i.e., that hold in all model expansions I_t of T with $I_p \subset I_t$ [1]. In the application, the propagation works interactively: as soon as a new rule is created, the impact on the eligibility is immediately shown by coloring the asset green (eligible) or red (not eligible). The *explanation* inference traces the propagated values back to the given values of the interpretation I_p [3]. The application allows the user to click on a propagated value and see immediately which atoms steered the decision. The theory comparison task uses the *model expansion* inference to compare two profiles. Active investors will typically update their profile regularly. In this case an automated comparison of two versions of the profile is helpful to ensure that correct amendments have been made. With the *model expansion* inference the logical equivalence of two theories can be checked by merging two theories T_1 and T_2, and adding the constraint that an asset can only be eligible in one of both theories. If no model I_t that satisfies T_3 is found, the two theories are equivalent.

Application development A prototype with a real-life example KB and the described inference tasks were showcased to the company in a prototype. Following this, the company has launched a project to further develop this prototype into a production application. The first technical release in production was done in February 2021 and a second release with improved workflow for signing the profiles between counterparties was released in June 2021. As of the second release, clients from large investment banks have access to a sandbox environment for training purposes. A full commercial roll-out will be done by September 2021. The target users for this commercial release are operations teams in the treasury back offices of large investment banks globally (target around 500 users across 150 organisations). The correctness of the knowledge base was insured by performing empirical tests with profile descriptions with up to 20 rules, and applied to portfolios of up to 300 assets with response times less than 3 seconds. These represent reasonable tranches for proper business use. Any larger portfolios can be tested off line with reporting being sent when processing has finished. Compared with the manual creation of a profile, the operational risk linked to the automation is almost non-existent thanks to the two-step procedure to turn natural language sentences via CNL automatically into an FO(.) KB. Once the KB is created, the application supports multiple services, such as the selection of eligible assets, optimisation of the associated costs and explanation of unexpected results.

Acknowledgements

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

1. Broes De Cat, Bart Bogaerts, M Bruynooghe, G Janssens, and Marc Denecker. Predicate logic as a modeling language: The idp system. In *Declarative Logic Programming: Theory, Systems, and Applications*, pages 279–329. ACM Books, 2018.
2. Marc Denecker. Building a knowledge base system for an integration of logic programming and classical logic. volume 5366, pages 71–76. Springer, 2008.
3. Marjolein Deryck, Jo Devriendt, Simon Marynissen, and Joost Vennekens. Legislation in the knowledge base paradigm: interactive decision enactment for registration duties. pages 174–177. IEEE, 2019.
4. Johan Wittocx, Maarten Mariën, and Marc Denecker. The idp system: a model expansion system for an extension of classical logic. In *Proceedings of the 2nd Workshop on Logic and Search*, pages 153–165. ACCO; Leuven, 2008.

Towards a Federated Fuzzy Learning System

Anna Wilbik[1][0000−0002−1989−0301] and Paul Grefen[2,3][0000−0003−2880−5110]

[1] Department of Data Science and Knowledge Engineering
Maastricht University, Maastricht, The Netherlands
a.wilbik@maastrichtuniversity.nl
[2] School of Industrial Engineering, Eindhoven University of Technology
Eindhoven, The Netherlands
[3] Atos Digital Transformation Consulting
Eindhoven, The Netherlands
P.W.P.J.Grefen@tue.nl

1 Introduction

The abundant availability of data allows the construction of predictive systems that support decision makers in business and society. A problem arises if an organization does not have a large enough data set by itself to construct a system of adequate quality. Obtaining additional data from other parties may be impossible because of competitive threats or privacy regulations, e.g. the EU General Data Protection Regulation (GDPR) [1].

To overcome these risks, federated learning is becoming increasingly popular to enable automated learning in distributed networks of autonomous partners without sharing raw data. Federated learning enables a collaboration between multiple parties to jointly train a machine learning model without exchanging the local data [7]. Because the data are not exchanged between parties, it is considered a privacy preserving approach. The collaboration in learning is considered successful, if for at least one party the performance of the federated model is better than the performance of the local model [5].

So far, only crisp systems have been used in this context. The use of a fuzzy inference system [8] can bring advantages to deal with vagueness and uncertainty in predictive systems. We show that it is indeed possible to build a fuzzy inference model in a federated learning setting, resulting in a Federated Fuzzy Learning System (F^{2}LS). We also show that this combination brings advantages to decision making that cannot be achieved with either mechanism in isolation.

2 Method: Constructing an F^{2}LS

The learning algorithm we use follows both the two-step process of training the Takagi-Sugeno fuzzy inference model [4, 6, 3] and the general federated learning process [7].

In the first stage of the algorithm (structure and rule antecedent identification), the server requests each client to cluster their local data and return to the server the cluster centers and the standard deviations. Next, similar clusters are

merged (i.e., cluster centers that are close enough are averaged). For this purpose we use agglomerative hierarchical clustering with a predefined threshold. In this process, two clusters from the same client cannot be merged. The number of merged clusters determines the number of rules in the F^{2}LS: for each cluster, one rule will be formed. The fuzzy sets in the rule antecedents are defined by the corresponding cluster as Gaussian membership with averaged cluster center $\bar{c}$ and averaged standard deviation $\bar{\sigma}$ as parameters.

In the second stage of the algorithm (rule consequent identification), we use the stochastic gradient descent algorithm in a federated setting. This means that each client selected in each round receives a federated model, runs E training passes of the stochastic gradient descent algorithm to find consequent parameters on a training batch of local data, and then returns the updated parameters to the server. The server updates the parameters of the rule consequent of the federated model as the weighted average of parameters returned by the clients in this round. The weights are dependent on the size of local data, such that large data sets have more influence than small data sets.

Details of the algorithm are described in our full paper [9].

3 Results: Testing an F^{2}LS

We have tested the proposed F^{2}LS on two small data sets from the UCI repository [2]. The goal of these experiments is to verify whether one can train a fuzzy inference system in a federated setting. As a success criterion we use the one proposed by Li et al. [5], in which a federated model should improve the performance for at least one party.

We have calculated MSE and MAE on the test sets available to each client, for both the local and federated models. Each experiment was repeated 20 times with random partitioning of the data. The mean of the errors shows that the federated learning setting is successful, as all parties on average improve their performance quality. However among the 20 repetitions, there are a few cases in which the federated model didn't outperform any of the local models. Further research is required to learn in which cases joining a federation is beneficial for a party.

4 Concluding remarks

We have proposed an approach for building an F^{2}LS, using a Takagi-Sugeno fuzzy inference system in a federated setting. The F^{2}LS approach integrates the best of two worlds: federated learning to deal with privacy-preserving data integration and learning and fuzzy inference to deal with uncertainty and vagueness in the contents of the learning process. We have shown that on average a federated model can outperform corresponding local models. The presented prototype approach requires further testing with an emphasis on test cases with heterogeneous data, as this is a known weak point of federated approaches.

References

1. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data (...) (general data protection regulation) oj l 119, 4.5.2016, p. 1–88
2. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
3. Herrera, F., Lozano, M., Verdegay, J.L.: Tuning fuzzy logic controllers by genetic algorithms. International Journal of Approximate Reasoning **12**(3-4), 299–315 (1995)
4. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-fuzzy and Soft Computing, a Computational Approach to Learning and Machine Intelligence (1997)
5. Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., He, B.: A survey on federated learning systems: vision, hype and reality for data privacy and protection. arXiv preprint arXiv:1907.09693 (2019)
6. Passino, K.M., Yurkovich, S.: Fuzzy control. MA: Addison-wesley (1998)
7. by: Peter Kairouz, E., McMahan, H.B.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1), – (2021). https://doi.org/10.1561/2200000083
8. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics (1), 116–132 (1985)
9. Wilbik, A., Grefen, P.: Towards a federated fuzzy learning system. In: 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). pp. 1–6. IEEE (2021)

RobBERT: a Dutch RoBERTa-based Language Model

Pieter Delobelle[1 [0000−0001−5911−5310]], Thomas Winters[1 [0000−0001−7494−2453]], and Bettina Berendt[1,2 [0000−0002−8003−3413]]

[1] Department of Computer Science, KU Leuven, Belgium; Leuven.ai
[2] Faculty of Electrical Engineering and Computer Science, TU Berlin, Germany
{firstname}.{lastname}@kuleuven.be

Abstract. BERT is a popular pre-trained language model used as a base for getting outstanding performance on a wide variety of natural language tasks.Recent studies show that BERT models trained on a single language significantly outperform the multilingual version. Also, its performance was improved by robustly optimising the architecture, as done in the RoBERTa model. We trained a Dutch language model called RobBERT and evaluated different tokenizers, its performance on various tasks and its fairness. The results show that it is a powerful pre-trained model for a large variety of Dutch language tasks, which we released to support further downstream Dutch NLP applications.

Keywords: Natural Language Processing · BERT model · RoBERTa

1 RobBERT

RobBERT is a pre-trained BERT-like Dutch language model, which can be used for various downstream natural language processing tasks. We trained it using the RoBERTa architecture and training regime [7], which optimised BERT's setting [6] e.g., by only using masked language modelling as pre-training task. The data used to train RobBERT is the Dutch part of the OSCAR corpus [9], based on Common Crawl where sentences were classified and split per language.

We evaluated RobBERT on several tasks against competing models, such as multilingual BERT (mBERT) [6] and other BERT models such as BERT-NL [3] and BERTje [4]. We also evaluated the importance of language-specific tokenizers by using the original (English) RoBERTa tokenizer (RobBERT v1) and training a new Dutch tokenizer (RobBERT v2). We evaluated the models on book review sentiment analysis (DBRD), *die/dat* co-reference resolution (die-dat) [1], part-of-speech tagging (POS) [10], We found that RobBERT outperforms the competing models on most tasks. This could be due to the fact that the RoBERTa architecture optimized the BERT architecture, and because RobBERT uses more data than its other Dutch counterparts. Another reason could be the

This extended abstract is the abbreviated version of the paper with the same name from *Findings of the Association for Computational Linguistics: EMNLP 2020* [5].

Table 1. Results on four benchmarks and comparison with other models, both Dutch and multilingual. *The dataset size for mBERT (indicated with †) is estimated using current Wikipedia dumps. Earlier reported results are annotated with their citations.*

	Pre-training data			Benchmark scores			
Model	**Datasets**	**Size**	**Vocab.**	**DBRD**	**DIE-DAT**	**NER**	**POS**
mBERT	Wikipedia	75† GB	WordPiece (int.)	—	98.3 ± 0.04	90.9 [11]	96.5 ± 0.3
BERT-NL	SoNaR [8]	2.2 GB	WordPiece	84.0 [3]	—	89.7 [3]	—
BERTje	SoNaR [8] + others	12 GB	WordPiece	93.0 [4]	98.3 ± 0.04	88.3 [4]	96.3 ± 0.3
RobBERT v1	OSCAR [9]	39 GB	BPE (En.)	94.4 ± 1.0	98.4 ± 0.04	87.5	96.4 ± 0.4
RobBERT v2	OSCAR [9]	39 GB	BPE	95.1 ± 0.9	99.2 ± 0.03	89.1	96.4 ± 0.4

nature of the training data, namely text scraped from the internet, as its stylistic diversity creates a more robust model.

2 Fairness

Since RobBERT is a model that could be used as a base model for a wide range of tasks, we evaluated its fairness by probing for gender biases. We did this by checking for gender stereotypes and its predictive performance on texts written by different genders. We translated an existing English dataset of professions [2] to Dutch, and filled these into three template sentences: a control template (‘‘`<mask> goes to a <T>.`’’) and two with co-reference (‘‘`<mask> is a <T>.`’’, ‘‘`<mask> works as a <T>`’’). We then checked how often the mask was filled in with *“he”* compared to *“she”*. For the co-referent templates, we found that RobBERT estimates the male pronoun more likely in almost all cases, even for processions with a gendered suffix. This is likely an artefact due to the male pronoun being much more present in text in general.

We also evaluated if RobBERT had unequal performance based on the writer of the review in the DBRD dataset. We augmented the test set with the gender of the writer who self-reported this on their user profile, and checked if our already fine-tuned model (which thus never saw this gender) had predictive parity for this sensitive attribute. Only 64% of users reported their gender, of which 76% were written by women. While the performance for just predicting positive reviews is about equal, we found that the finetuned model has a higher performance for predicting highly positive reviews when written by women than by men.

3 Conclusion

We introduced a new pre-trained Dutch language model called RobBERT, and showed that it outperforms earlier approaches on various language tasks. The RobBERT model can thus serve as a useful base that can be fine-tuned on new datasets, and thus help foster new models that advance results for a diverse range of Dutch language tasks.

References

1. Allein, L., Leeuwenberg, A., Moens, M.F.: Binary and Multitask Classification Model for Dutch Anaphora Resolution: Die/Dat Prediction. arXiv:2001.02943 [cs] (Jan 2020), http://arxiv.org/abs/2001.02943
2. Bolukbasi, T., Chang, K.W., Zou, J.Y., Saligrama, V., Kalai, A.T.: Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In: Advances in Neural Information Processing Systems. pp. 4349–4357 (2016)
3. Brandsen, A., Dirkson, A., Verberne, S., Sappelli, M., Manh Chu, D., Stoutjesdijk, K.: BERT-NL a set of language models pre-trained on the Dutch SoNaR corpus (Nov 2019), http://textdata.nl
4. de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., Nissim, M.: BERTje: A Dutch BERT Model. arXiv:1912.09582 [cs] (Dec 2019), http://arxiv.org/abs/1912.09582
5. Delobelle, P., Winters, T., Berendt, B.: RobBERT: a Dutch RoBERTa-based Language Model. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 3255–3265. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.292, https://aclanthology.org/2020.findings-emnlp.292
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423
7. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs] (Jul 2019), http://arxiv.org/abs/1907.11692
8. Oostdijk, N., Reynaert, M., Hoste, V., Schuurman, I.: The construction of a 500-million-word reference corpus of contemporary written dutch. In: Essential speech and language technology for Dutch, pp. 219–247. Springer (2013)
9. Ortiz Suárez, P.J., Sagot, B., Romary, L.: Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In: 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7). Cardiff, United Kingdom (Jul 2019), https://hal.inria.fr/hal-02148693
10. Van Noord, G., Bouma, G., Van Eynde, F., De Kok, D., Van der Linde, J., Schuurman, I., Sang, E.T.K., Vandeghinste, V.: Large scale syntactic annotation of written Dutch: Lassy, pp. 147–164. Springer (2013)
11. Wu, S., Dredze, M.: Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 833–844. Association for Computational Linguistics, Hong Kong, China (2019). https://doi.org/10.18653/v1/D19-1077

A Note on Pattern Classification with Evolving Long-term Cognitive Networks

Gonzalo Nápoles[1], Agnieszka Jastrzębska[2], and Yamisleydi Salgueiro[3]

[1] Department of Cognitive Science & Artificial Intelligence, Tilburg University, The Netherlands g.r.napoles@uvt.nl
[2] Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland
[3] Department of Computer Science, Faculty of Engineering, Universidad de Talca, Campus Curicó, Chile

1 Evolving Long-term Cognitive Network

In the discussed journal paper, we proposed an interpretable neural system for data classification termed Evolving Long-term Cognitive Network (ELTCN). The ELTCN model builds upon the Long-term Cognitive Network (LTCN) [3], but what makes it distinct is that it allows for the weights to change from an iteration to another during the reasoning process. The ELTCN is a neural architecture with two layers. We envisioned that the Fuzzy Cognitive Map (FCM) [1] model would be embedded in this architecture as an input layer. The FCM can get unfolded without losing the ability to interpret the nodes and the weights in the resulting architecture. The second layer is the output layer.

Let us introduce the backbone of data processing of an unfolded ELTCN with T abstract layers, each containing M neurons. We have N output neurons and $P = M + N$. Let $w_{ji}^{(t)}$ be a weight in the t-th iteration and $a_i^{(t)}$ the activation value of the i-th neuron. Eq. (1) shows how to compute neurons' activation values by following the evolving reasoning principle,

$$a_i^{(t+1)} = f_i^{(t+1)} \left(\sum_{j=1}^{P} w_{ji}^{(t)} a_j^{(t)} \right) \tag{1}$$

where $f_i^{(t+1)}(x)$ can be either the sigmoid function,

$$s_i^{(t)}(x) = \frac{1}{1 + e^{-\lambda_i^{(t)}(x - h_i^{(t)})}} \tag{2}$$

or the hyperbolic tangent function,

$$q_i^{(t)}(x) = \frac{e^{2\lambda_i^{(t)}(x - h_i^{(t)})} - 1}{e^{2\lambda_i^{(t)}(x - h_i^{(t)})} + 1} \tag{3}$$

where $\lambda_i^{(t)} > 0$ and $h_i^{(t)} \in \mathbb{R}$ denote the function slope and its offset, respectively. The activation values for output neurons is given by:

$$a_i^{(t+1)} = \frac{e^{\left(\sum_{j=1}^{M} w_{ji}^{(t)} a_j^{(t)}\right)}}{e^{\left(\sum_{k=1}^{N}\left(\sum_{j=1}^{M} w_{jk}^{(t)} a_j^{(t)}\right)\right)}}. \tag{4}$$

An essential contribution of this paper was related to a new backpropagation algorithm that implements the ELTCN reasoning process and adjusts the weights of this neural model and some transfer function parameters.

The learning algorithm includes two regularization components that attempt to produce neural models we can understand. They allow minimizing the weight variability between two consecutive iterations and the offset values. Producing "stable" weights makes the model easier to interpret. We also modified the weight normalization procedure presented in [2] to this model.

2 Results and Conclusion

In the paper, we demonstrated the predictive power of the proposed model in a series of numerical simulations concerning 58 pattern classification datasets. We also showed how to derive intrinsic explanations. The state-of-the-art classifiers selected for the comparative analysis were as follows: Logistic Regression (LR), Gaussian Naive Bayes (GNB), Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), and Multilayer Perceptron (MLP). We made a motivated decision of not optimizing the hyperparameters of neither of the algorithms used, including the ELTCN.

The key findings were that the proposed ELTCN model (together with the new backpropagation learning method) attains competitive prediction rates concerning traditional classifiers. Fig. 1 presents the average Kappa value achieved by each classification model after performing 10-fold cross-validation. It can be observed that MLP, RF, and ELTCN report the highest prediction rates in this study but only ELTCN provides intrinsic interpretability.

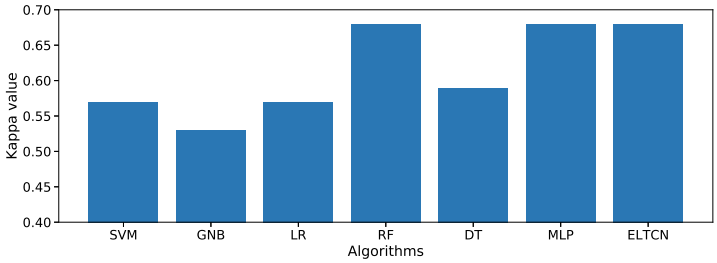


Fig. 1. Average Kappa value achieved by each classifier on the 58 datasets.

Furthermore, we observed that the ELTCN variant using the hyperbolic tangent function is more accurate in terms of Kappa values than the variant using the sigmoid function when it comes to the Kappa values.

References

1. Kosko, B.: Fuzzy cognitive maps. International Journal of Man-Machine Studies **24**(1), 65 – 75 (1986)
2. Nápoles, G., Jastrzębska, A., Mosquera, C., Vanhoof, K., Homenda, W.: Deterministic learning of hybrid fuzzy cognitive maps and network reduction approaches. Neural Networks **124**, 258–268 (2020)
3. Nápoles, G., Vanhoenshoven, F., Falcon, R., Vanhoof, K.: Nonsynaptic error backpropagation in long-term cognitive networks. IEEE Transactions on Neural Networks and Learning Systems **31**, 865–875 (2019)

SAGE: Intrusion Alert-driven Attack Graph Extractor (Encore abstract)

Azqa Nadeem[1], Sicco Verwer[1], Stephen Moskal[2], and Shanchieh Jay Yang[2]

[1] Delft University of Technology, The Netherlands
{azqa.nadeem,s.e.verwer}@tudelft.nl
[2] Rochester Institute of Technology, USA
{sfm5015,jay.yang}@rit.edu

Abstract

Security Operations Center (SOC) analysts investigate thousands of intrusion alerts on a daily basis, leading to alert fatigue and reduced productivity [1]. While alert correlation techniques help reduce the volume of alerts, they do not show the bigger picture of how the attack happened. Attack graphs (AG) are visual models of attacker strategies. State-of-the-art approaches for AG generation focus mostly on deriving dependencies between system vulnerabilities, based on network scans and expert knowledge [3]. In real-world operations however, it is costly and ineffective to rely on constant vulnerability scanning and expert-crafted AGs.

We propose to learn AGs, purely based on the actions observed through intrusion alerts. In this paper, we develop an unsupervised sequence learning system, called SAGE (Intru**S**ion alert-driven **A**ttack **G**raph **E**xtractor)[3]. It constructs alert-driven AGs without any expert input. These AGs unlock a new means to derive intelligence regarding attacker strategies without having to investigate thousands of intrusion alerts.

Class imbalance remains a major challenge for machine learning-enabled attacker strategy identification – severe alerts are infrequent, while non-severe alerts (related to network scans) are very frequent. This makes most machine learning solutions inherently unsuitable, since they discard infrequent behavior. Instead, we learn an interpretable suffix-based probabilistic deterministic finite automaton (S-PDFA) using the FlexFringe automaton learning framework [4]. We tune the learning algorithm and transform the alert data such that the resulting model accentuates infrequent severe alerts, without discarding any low-severity alerts. The model summarizes attack paths leading to severe attack stages. It can distinguish between alerts with the same signature but different contexts, i.e., scanning at the start and scanning midway through an attack are treated differently, since they indicate different attack stages. Targeted attack graphs are extracted from the S-PDFA on a per-victim, per-objective basis.

Tested with intrusion alerts collected through Collegiate Penetration Testing Competition [2], we evaluate SAGE's efficacy on distributed, multi-stage attack

[3] SAGE is open-source: https://github.com/tudelft-cda-lab/SAGE

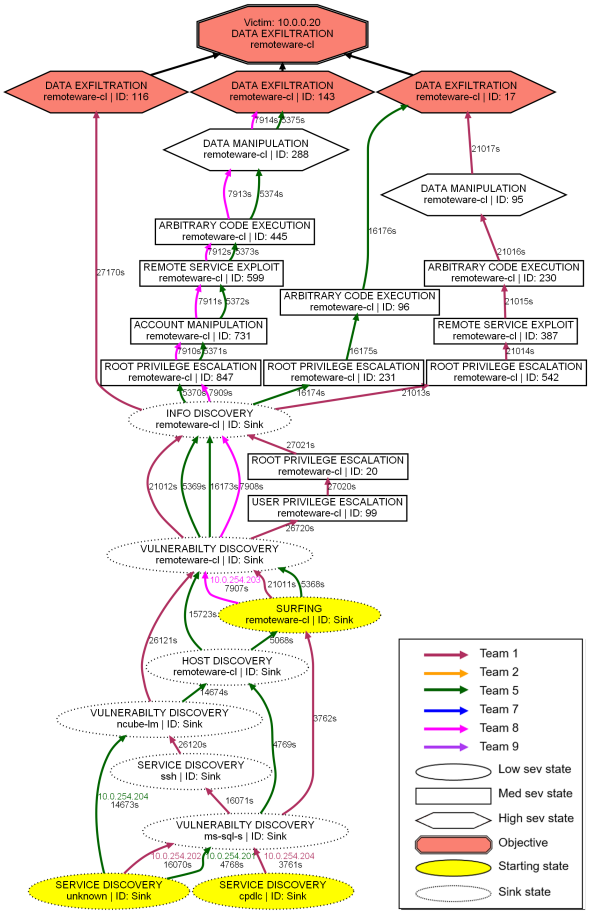


Fig. 1. Alert-driven attack graph of data exfiltration (IDs are state identifiers). The S-PDFA finds 3 *ways of exploiting* the objective based on path differences.

scenarios. SAGE compresses over 330k alerts into just 93 AGs, while also showing how a specific attack transpired. For instance, Fig. 1 shows 3 teams conducting data exfiltration. The AGs capture the strategies used by the participating teams, producing directly relevant insights for SOC analysts, *e.g.,* they reveal that attackers follow shorter paths after they have discovered a longer one. In Fig. 1, Teams 1 and 5 make two attempts, where each subsequent attempt is shorter than the first. This happens in 84.5% of the cases. They also provide an intuitive layout to compare attacker strategies for discovering parallel attacks and fingerprintable paths. We believe that alert-driven attack graphs can play a key role in AI-enabled cyber threat intelligence as they open up new avenues for attacker strategy analysis whilst reducing analysts' workload.

References

1. Hassan, W.U., Guo, S., Li, D., Chen, Z., Jee, K., Li, Z., Bates, A.: Nodoze: Combatting threat alert fatigue with automated provenance triage. In: NDSS (2019)
2. Munaiah, N., Rahman, A., Pelletier, J., Williams, L., Meneely, A.: Characterizing attacker behavior in a cybersecurity penetration testing competition. In: 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). pp. 1–6. IEEE (2019)
3. Noel, S., Elder, M., Jajodia, S., Kalapa, P., O'Hare, S., Prole, K.: Advances in topological vulnerability analysis. In: CATCH. pp. 124–129. IEEE (2009)
4. Verwer, S., Hammerschmidt, C.A.: Flexfringe: a passive automaton learning package. In: 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME). pp. 638–642. IEEE (2017)

Everyone knows that everyone knows (abstract)

Hans van Ditmarsch[1], Malvin Gattinger[2], and Rahim Ramezanian[3]

[1] Open University, the Netherlands
hans.vanditmarsch@ou.nl
[2] University of Amsterdam, the Netherlands
malvin@w4eg.eu
[3] Shomara LLC, Tehran, Iran
rahim.ramezanian@gmail.com

Extended abstract of work published as [17] and available in revised extended version as [10]. Verifications were assisted by the model checker GoMoChe available at https://github.com/m4lvin/GoMoChe.

The gossip problem addresses how to spread secrets among a group of agents by pairwise message exchanges: telephone calls. Each agent holds a single secret, and when calling each other the agents exchange all the secrets they know. An agent may call another agent if it has that agent's telephone number. The goal of the information dissemination is that all agents know all secrets. The situation can be represented by a network where the nodes are the agents and where, when two nodes are linked, the agents can call each other.

There are many variations of the problem. It goes back to the early 1970s [5, 18, 16, 6, 19, 13]. In this classic setting only secrets are exchanged, and the focus is on minimum execution length of protocols executed by a central scheduler. Later publications assume that the scheduling is *distributed* [15, 12]. Fairly recent developments focus on gossip protocols with *epistemic* preconditions for calls [1, 3, 4, 2, 7, 11]. For example, agents may only call another agent once, or only if they do not know the other agent's secret, etc. In *dynamic* gossip [8, 9] the agents do not only exchange all the secrets they know but also all the telephone numbers they know. This results in network expansion: not only the secret relation but also the number relation is expanded after a call. A way to load the messages beyond merely exchanging secrets is to exchange *knowledge about secrets* [14]. One can thus achieve higher-order shared knowledge of all secrets (all the agents know that all the agents know, etc.).

In this contribution we investigate gossip protocols with the epistemic goal that all agents know that all agents know all secrets. Unlike [14] we continue to assume that agents only exchange secrets. However, we additionally assume that the agents may have knowledge of the protocol, where we consider four well-investigated gossip protocols, and we also model additional behaviour of agents, and how they affect properties such as termination and execution length. The following summarize our approach:

- The protocol terminates if *everyone knows that everyone knows all secrets.*
- Agents *know the gossip protocol* that is used by all agents.
- Agents who know that everyone knows all secrets *no longer make calls.*

– Agents who know that everyone knows all secrets *no longer answer calls.*

An agent who knows all secrets is an *expert*. An agent who knows that everyone is an expert is a *super expert*. So our epistemic goal is for all agents to become super experts, where we also investigate the effect of additional assumptions that the protocol is known and that super experts no longer make and answer calls. Asynchronous conditions where agents are only aware of calls involving them, are distinguished from synchronous conditions where agents are aware of calls taking place but not who make them if they are not involved.

Below is a simple example for four agents a, b, c, d under asynchronous considitions. The rows describe the effect of successive calls. The columns describe what respectively a, b, c, d know: a lower case y in the column of agent x means that x knows the secret of y; an upper case Y means that x knows that y knows all secrets. Therefore, "abcd" denotes an expert and "ABCD" denotes a super expert.

	a	b	c	d	all only know own secret
$\overset{ab}{\rightarrow}$	ab	ab	c	d	
$\overset{cd}{\rightarrow}$	ab	ab	cd	cd	
$\overset{ac}{\rightarrow}$	abcd A C	ab	abcd A C	cd	
$\overset{bd}{\rightarrow}$	abcd A C	abcd B D	abcd A C	abcd B D	
$\overset{ab}{\rightarrow}$	abcd ABC	abcd AB D	abcd A C	abcd B D	
$\overset{ad}{\rightarrow}$	abcd ABCD	abcd AB D	abcd A C	abcd AB D	a is a super expert
$\overset{bc}{\rightarrow}$	abcd ABCD	abcd ABCD	abcd ABC	abcd AB D	b is a super expert
$\overset{cd}{\rightarrow}$	abcd ABCD	abcd ABCD	abcd ABCD	abcd ABCD	all are super experts

We present a logical language and semantics for gossip protocols with the epistemic goal that all agents know that all agents know all secrets. A protocol is super-successful if all executions terminate satisfying this condition. We recall four gossip protocols from the literature: ANY, PIG, CMO, and LNS. We obtain various results for the protocols ANY and PIG, mainly that they are super-successful (both for the synchronous and asynchronous versions) in some sense adequate for protocols permitting infinite call sequences. We further refine the logic in order to model common knowledge of gossip protocols. If a protocol is common knowledge we call it a 'known protocol'. We then show that synchronous known CMO is super-successful. Subsequently we refine the semantics of commonly known protocols with the feature that super experts do not make calls and do not answer calls. We then show that, if this is also known, super-successful protocol executions can be shorter. However, under these conditions CMO is no longer super-successful. An even further refinement of the semantics adds 'skip calls' following terminal protocol-permitted sequences, that allow us to regain a super-successful CMO, and that seems a promising feature to adapt or repair yet other epistemic gossip protocols.

References

1. Apt, K., Grossi, D., van der Hoek, W.: Epistemic protocols for distributed gossiping. In: Proceedings of 15th TARK. pp. 51–66 (2015). https://doi.org/10.4204/EPTCS.215.5
2. Apt, K., Wojtczak, D.: Verification of distributed epistemic gossip protocols. J. Artif. Intell. Res. **62**, 101–132 (2018). https://doi.org/10.1613/jair.1.11204
3. Attamah, M., van Ditmarsch, H., Grossi, D., van der Hoek, W.: Knowledge and gossip. In: Proc. of 21st ECAI. pp. 21–26. IOS Press (2014). https://doi.org/10.3233/978-1-61499-419-0-21
4. Attamah, M., van Ditmarsch, H., Grossi, D., van der Hoek, W.: The pleasure of gossip. In: Başkent, C., Moss, L., Ramanujam, R. (eds.) Rohit Parikh on Logic, Language and Society. pp. 145–163. Springer (2017)
5. Baker, B., Shostak, R.: Gossips and telephones. Discrete Mathematics **2**(3), 191–193 (1972). https://doi.org/10.1016/0012-365X(72)90001-5
6. Boyd, D., Steele, J.: Random exchanges of information. Journal of Applied Probability **16**, 657–661 (1979). https://doi.org/10.2307/3213094
7. Cooper, M., Herzig, A., Maffre, F., Maris, F., Régnier, P.: The epistemic gossip problem. Discret. Math. **342**(3), 654–663 (2019). https://doi.org/10.1016/j.disc.2018.10.041
8. van Ditmarsch, H., van Eijck, J., Pardo, P., Ramezanian, R., Schwarzentruber, F.: Epistemic protocols for dynamic gossip. J. Applied Logic **20**, 1–31 (2017). https://doi.org/10.1016/j.jal.2016.12.001
9. van Ditmarsch, H., van Eijck, J., Pardo, P., Ramezanian, R., Schwarzentruber, F.: Dynamic gossip. Bulletin of the Iranian Mathematical Society **45(3)**, 701–728 (2019). https://doi.org/10.1007/s41980-018-0160-4, https://arxiv.org/abs/1511.00867
10. van Ditmarsch, H., Gattinger, M., Ramezanian, R.: Everyone knows that everyone knows (2021), https://arxiv.org/abs/2011.13203
11. van Ditmarsch, H., van der Hoek, W., Kuijer, L.: The logic of gossiping. Artificial Intelligence **286**, 103306 (2020). https://doi.org/10.1016/j.artint.2020.103306
12. Eugster, P., Guerraoui, R., Kermarrec, A., Massoulié, L.: Epidemic information dissemination in distributed systems. IEEE Computer **37**(5), 60–67 (2004). https://doi.org/10.1109/MC.2004.1297243
13. Hedetniemi, S., Hedetniemi, S., Liestman, A.: A survey of gossiping and broadcasting in communication networks. Networks **18**, 319–349 (1988). https://doi.org/10.1002/net.3230180406
14. Herzig, A., Maffre, F.: How to share knowledge by gossiping. AI Commun. **30**(1), 1–17 (2017). https://doi.org/10.3233/AIC-170723
15. Kermarrec, A.M., van Steen, M.: Gossiping in distributed systems. SIGOPS Oper. Syst. Rev. **41**(5), 2–7 (2007). https://doi.org/10.1145/1317379.1317381
16. Knödel, W.: New gossips and telephones. Discrete Mathematics **13**, 95 (1975)
17. Ramezanian, R., Ramezanian, R., Gattinger, M., van Ditmarsch, H.: Everyone knows that everyone knows. In: Mojtahedi, M., Rahman, S., Zarepour, M. (eds.) Mathematics, Logic, and Their Philosophies: Essays in Honour of Mohammad Ardeshir. pp. 117–133. Springer (2021). https://doi.org/10.1007/978-3-030-53654-1
18. Tijdeman, R.: On a telephone problem. Nieuw Archief voor Wiskunde **3(19)**, 188–192 (1971)
19. West, D.: A class of solutions to the gossip problem, part I. Discrete Mathematics **39**(3), 307–326 (1982)

Deep tree-ensembles for multi-output prediction

Felipe Kenji Nakano[1,2], Konstantinos Pliakos[1,2], and Celine Vens[1,2]

[1] KU Leuven, Campus KULAK, Dept. of Public Health and Primary Care, Kortrijk, Belgium
[2] Itec, imec research group at KU Leuven, Kortrijk, Belgium
{felipekenji.nakano,konstantinos.pliakos,celine.vens}@kuleuven.be

Abstract. Recently, deep neural networks have expanded the state-of-art in various scientific fields and provided solutions to long standing problems across multiple application domains. Nevertheless, they also suffer from weaknesses since their optimal performance depends on massive amounts of training data and the tuning of an extended number of parameters. As a countermeasure, some deep-forest methods have been recently proposed, as efficient and low-scale solutions. Despite that, these approaches simply employ label classification probabilities as induced features and primarily focus on traditional classification and regression tasks, leaving multi-output prediction under-explored. Moreover, recent work has demonstrated that tree-embeddings are highly representative, especially in structured output prediction. In this direction, we propose a novel deep tree-ensemble (DTE) model, where every layer enriches the original feature set with a representation learning component based on tree-embeddings. In this paper, we specifically focus on two structured output prediction tasks, namely multi-label classification and multi-target regression. We conducted experiments using multiple benchmark datasets and the obtained results confirm that our method provides superior results to state-of-the-art methods in both tasks.

Keywords: · Deep-Forest · Multi-output prediction

1 Introduction and Method

Recently, deep learning has arisen as a cutting edge methodology advancing the state-of-the-art in many domains. Apart from its success, deep neural networks suffer also from weaknesses. For example, their training is computationally very expensive and demands large-scale datasets. As an alternative, deep-forest methods have been recently investigated as efficient and low-scale solutions [7, 6].

Motivated by them, we proposed a novel deep tree-ensemble (DTE) method where deep-models are built by sequentially concatenating layers of ensemble models. Here, we present a condensed version of our work, while a detailed description of our method and evaluation setup is available in the main publication [4]. In every layer of our model, we include a tree-based representation learning step extending the original input space with low-dimensional tree-embeddings (TEs) [5]. Different from [7, 6], instead of simply using the predictions of the

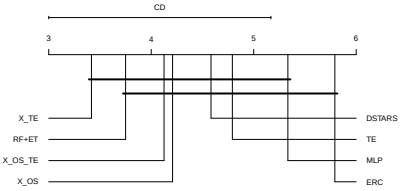
previous layer as extra attributes for the next one, we enrich the input feature space with a representation learning component based on the decision paths of the trees in the tree-ensemble collection. Our main hypothesis is that adding a representation learning component to the deep forest methodology will boost predictive performance while keeping computational complexity low.

TEs are generated as follows: All trees in the ensemble are converted to a binary vector $C = \{c_1, ...c_{|C|}\}$, where $|C|$ is the total number of nodes. Each $c_i \in C$ is treated as a feature, creating a representation $F \in R^{|N| \times |C|}$, where N stands for the number of instances in the dataset. $F_{ij} = 1$ if an instance belongs to such node, otherwise $F_{ij} = 0$. F is then filtered and weighted based on how frequently a node is traversed and PCA is used to generate the final embeddings.

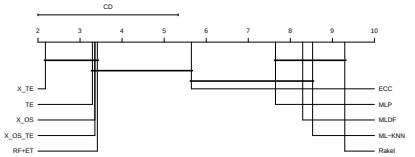
2 Experiments

We have focused on two multi-output prediction tasks: multi-label classification (MLC) and multi-target regression (MTR). For both tasks, we have collected multiple datasets from public repositories and performed 5-fold cross-validation.

To summarize our experiments, we present the Friedman-Nemenyi diagrams (Figures 1a and 1b). We have compared 4 variants of our method, each variant is described by the representation used at each layer. In this case, X stands for the original features, TEs for the tree-embeddings and OS for output space features.



(a) Diagram using aRRMSE



(b) Diagram using Label-ranking

As can be seen, our variant X_TE is ranked higher than its main competitor methods DSTARS and MLDF in MTR and MLC, respectively [2, 6]. Moreover, X_TE being ranked first states that TEs should be employed with the original representation (X) and not replace it, as the TE variant performed worse than X_TE. Additionally, the variants with OS are constantly ranked lower than its counterparts with TEs, reinforcing the representational power of the latter.

3 Conclusion

In this paper, we have proposed a novel deep tree-ensemble model for multi-output prediction tasks which integrates tree-embeddings. Our experiments have shown that the proposed model yields superior results in both tasks.

In future work, we would like to investigate tree-embeddings in tasks with a larger number of labels, such as hierarchical multi-label classification [3] and extreme multi-label classification [1].

References

1. Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in neural information processing systems*, pages 730–738, 2015.
2. Saulo Martiello Mastelini, Everton Jose Santana, Ricardo Cerri, and Sylvio Barbon. Dstars: A multi-target deep structure for tracking asynchronous regressor stacking. *Applied Soft Computing*, 91:106215, 2020.
3. Felipe Kenji Nakano, Mathias Lietaert, and Celine Vens. Machine learning for discovering missing or wrong protein function annotations. *BMC bioinformatics*, 20(1):485, 2019.
4. Felipe Kenji Nakano, Konstantinos Pliakos, and Celine Vens. Deep tree-ensembles for multi-output prediction. *Pattern Recognition*, page 108211, 2021.
5. Konstantinos Pliakos and Celine Vens. Mining features for biomedical data using clustering tree ensembles. *Journal of biomedical informatics*, 85:40–48, 2018.
6. Liang Yang, Xi-Zhu Wu, Yuan Jiang, and Zhi-Hua Zhou. Multi-label learning with deep forest. In *24th European Conference on Artificial Intelligence (ECAI' 20)*, pages 1634–1641, Santigo de Compostela,Spain, 2020.
7. Zhi-Hua Zhou and Ji Feng. Deep forest. *National Science Review*, 6(1):74–86, 2019.

Prompt Tuning or Fine-Tuning - Investigating Relational Knowledge in Pre-Trained Language Models

Leandra Fichtel[1][0000−0002−2696−169X], Jan-Christoph Kalo[2][0000−0002−5492−2292], and Wolf-Tilo Balke[1][0000−0002−5443−1215]

[1] Institute for Information Systems, TU Braunschweig
l.fichtel@tu-bs.de
balke@ifis.cs.tu-bs.de
[2] Knowledge Representation and Reasoning Group, VU Amsterdam
j.c.kalo@vu.nl

This is an extended abstract describing the paper "Prompt Tuning or Fine-Tuning - Investigating Relational Knowledge in Pre-Trained Language Models" published at AKBC 2021 [2].

Keywords: Language Models · Knowledge Graphs · Knowledge Graph Construction

1 Introduction

Recent research has shown that large pre-trained language models may serve as rich sources of relational knowledge [6]. The paper *Language Models as Knowledge Bases* has shown that it is possible to extract relational knowledge from arbitrary masked language models by completing cloze-style sentences. As an example, the sentence *Albert Einstein was born in [MASK]* could be completed by the word *Ulm*. Using this idea, factual knowledge from Wikidata [8] can be extracted from the language model BERT with a precision of around 31%.

The quality of these extractions, however, is strongly dependent on the formulation of the input sentence, the so-called *prompt*. A lot of effort was invested into tuning the prompt by complex mining and learning techniques, such that the extraction quality is improved up to 43% without changing the underlying language model [7, 1, 4, 3]. However, existing techniques usually need a significant amount of training data in the form of existing knowledge graph triples and a large amount of training time to optimize these prompts by using complex additional models.

In our paper, we perform a simple adaptive fine-tuning method on the original language model using only a few training triples from an existing knowledge graph. We continue training on the pre-training objective using masked sentences: Given the triple (Albert Einstein, bornIn, Ulm), we use the sentence *Albert Einstein was born in [MASK]* and the correct answer *Ulm* as a training input. By this very simple idea, we achieve superior results on the LAMA probe with a precision of over 48% without the need of a complex prompt tuning technique.

2 Experiments and Results

In this work, we perform three experiments to evaluate our adaptive fine-tuning model (BERTriple) using the standard LAMA benchmark dataset ([6]) for fact extraction from language models.

In the first experiment, we show that BERTriple can significantly improve upon the best state-of-the-art baselines on the LAMA probe as depicted in Table 1. Our method outperforms existing methods by around 5% precision if we use at most 1000 training triples for adaptive fine-tuning.

Table 1. P@1 [%] of four baselines and our model BERTriple evaluated on LAMA

Test Dataset	BERT	LPAQA	BERTese	AutoPrompt	**BERTriple**
LAMA	31.1	34.1	38.3	43.3	**48.4**

In the second experiment, we investigate how prone our training approach is to using smaller training datasets. We evaluate the precision of different training dataset sizes on LAMA. Our method still achieves 45% precision With only 50 training triples per relation. Thus it outperforms the prompting methods with a very small amount of training data.

The third experiment evaluates the transfer learning capabilities of our model BERTriple by leaving out the training data for single relationships from the training procedure. Considering the precision of the models, the relations can be clustered into three groups. The precision is either (a) in the same range of the original BERT, (b) better than BERT and in the same range of our method BERTriple (bold), (c) or notably lower than BERT. Hence, some relationships show good transfer learning capabilities and even improve when not trained on, while others show not transfer learning capabilities at all.

3 Conclusions

There is one major difference between prompt tuning techniques and adaptive fine-tuning. Whereas the main goal of prompt tuning is to use the prompts for many downstream tasks and not to save separate language models for each task [5], adaptive fine-tuning creates a model which is limited to the cloze-style fact extraction task. For a different task, a new adaptive fine-tuning has to be executed. However, as discussed in this work, most models for prompt tuning are complex and add a significant extra training effort, even though using tuned prompts results in a worse fact extraction performance in contrast to our adaptive fine-tuning. Consequently, instead of reaching the goal to have a single solution for all tasks, fine-tuning a pre-trained language model offers a more computational efficient solution to achieve superior fact extraction performance.

References

1. Bouraoui, Z., Camacho-Collados, J., Schockaert, S.: Inducing relational knowledge from bert. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 7456–7463 (Apr 2020). https://doi.org/10.1609/aaai.v34i05.6242, https://ojs.aaai.org/index.php/AAAI/article/view/6242
2. Fichtel, L., Kalo, J.C., Balke, W.T.: Prompt tuning or fine-tuning - investigating relational knowledge in pre-trained language models. In: Automatic Knowledge Base Construction (2021), https://openreview.net/forum?id=o7sMlpr9yBW
3. Haviv, A., Berant, J., Globerson, A.: BERTese: Learning to speak to BERT. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. pp. 3618–3623. Association for Computational Linguistics, Online (Apr 2021), https://www.aclweb.org/anthology/2021.eacl-main.316
4. Jiang, Z., Xu, F.F., Araki, J., Neubig, G.: How can we know what language models know? Transactions of the Association for Computational Linguistics **8**, 423–438 (2020). https://doi.org/10.1162/tacl_a_00324, https://www.aclweb.org/anthology/2020.tacl-1.28
5. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning (2021)
6. Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.: Language models as knowledge bases? In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2463–2473. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1250, https://www.aclweb.org/anthology/D19-1250
7. Shin, T., Razeghi, Y., Logan IV, R.L., Wallace, E., Singh, S.: AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4222–4235. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.346, https://www.aclweb.org/anthology/2020.emnlp-main.346
8. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)

Attention is not *all* you need: pure attention loses rank doubly exponentially with depth

Yihe Dong*[1], Jean-Baptiste Cordonnier[2], and Andreas Loukas*[2]

[1] Google
[2] EPFL

Abstract. We propose a new way to understand self-attention networks [1,4]: we prove that self-attention possesses a strong inductive bias towards "token uniformity". Specifically, without skip connections, the output converges doubly exponentially to a rank-1 matrix. On the other hand, skip connections stop the output from degeneration. Our experiments verify the convergence results on a standard transformer architecture.

1 Attention loses rank doubly exponentially

We start by providing background in Sec. 1.1 on self-attention networks (SANs). Sec. 1.2 then proves that SANs converge doubly exponentially (with depth) to a rank-1 matrix. The role of skip connections is studied in Sec 1.3. Finally, Sec 1.4 empirically investigates rank collapse in a real architecture.

1.1 Background: self-attention networks

Let $\boldsymbol{X}$ be a $n \times d_{in}$ input tensor consisting of n tokens. An SAN is built out of L multi-head self-attention layers, each having H heads. The output of the h-th self-attention head can be written as $\mathrm{SA}_h(\boldsymbol{X}) = \boldsymbol{P}_h \boldsymbol{X} \boldsymbol{W}_{V,h} + \mathbf{1}\boldsymbol{b}_{V,h}^\top$, where $\boldsymbol{W}_{V,h}$ is a $d_{in} \times d_v$ value weight matrix and the $n \times n$ *row-stochastic* matrix $\boldsymbol{P}_h$ is given by $\boldsymbol{P}_h = \mathrm{softmax}(d_{qk}^{-\frac{1}{2}}(\boldsymbol{X}\boldsymbol{W}_{QK,h}\boldsymbol{X}^\top + \mathbf{1}\boldsymbol{b}_{Q,h}^\top \boldsymbol{W}_{K,h}^\top \boldsymbol{X}^\top))$, where the key and query weight matrices $\boldsymbol{W}_{K,h}$ and $\boldsymbol{W}_{Q,h}$ are of size $d_{in} \times d_{qk}$, whereas $\boldsymbol{W}_{QK,h} = \boldsymbol{W}_{Q,h}\boldsymbol{W}_{K,h}^\top$. The output of each SAN layer is $\mathrm{SA}(\boldsymbol{X}) = \mathbf{1}[\boldsymbol{b}_{O,1}^\top, \cdots, \boldsymbol{b}_{O,H}^\top] + [\mathrm{SA}_1(\boldsymbol{X}), \cdots, \mathrm{SA}_H(\boldsymbol{X})]\ [\boldsymbol{W}_{O,1}^\top, \cdots, \boldsymbol{W}_{O,H}^\top]^\top$ where we set $\boldsymbol{W}_h = \boldsymbol{W}_{V,h}\,\boldsymbol{W}_{O,h}^\top$ and $\boldsymbol{b}_O = \sum_h \boldsymbol{b}_{O,h}$ and $[H] = [1, \ldots, H]$. Let $\boldsymbol{X}^l$ be the output of the l-th layer and fix $\boldsymbol{X}^0 = \boldsymbol{X}$. As is common practice, we let all layers consist of the same number of heads. The SAN output is given by repeating the recursion $\boldsymbol{X}^l = \mathrm{SA}^l(\boldsymbol{X}^{L-1})$ over $l \in [L]$ layers.

1.2 The rank collapse phenomenon

We now move on to analyze the convergence of deep SANs with multiple heads per layer. We examine, in particular, how the residual

$$\mathrm{res}(\boldsymbol{X}) = \boldsymbol{X} - \mathbf{1}\boldsymbol{x}^\top, \quad \text{where} \quad \boldsymbol{x} = \mathrm{argmin}_{\boldsymbol{x}} \|\boldsymbol{X} - \mathbf{1}\boldsymbol{x}^\top\|$$

changes during the forward pass. We also denote the ℓ_1, ℓ_∞-composite norm of a matrix $\boldsymbol{X}$ as $\|\boldsymbol{X}\|_{1,\infty} = \sqrt{\|\boldsymbol{X}\|_1 \|\boldsymbol{X}\|_\infty}$. We note that $\ell_{1,\infty}$ is not a proper norm as it does not satisfy the triangle inequality, though it is absolutely homogeneous and positive definite. Our main result is as follows:

* Yihe Dong and Andreas Loukas contributed equally to the full version of this article which appeared as an oral contribution to ICML 2021.

Theorem 1. *In a depth-L and width-H SAN without skip connections, let $\|\boldsymbol{W}^l_{QK,h}\|_1\|\boldsymbol{W}^l_h\|_{1,\infty} \leq \beta$ for all heads $h \in [H]$ and layers $l \in [L]$, then:*

$$\|res(SAN(\boldsymbol{X}))\|_{1,\infty} \leq \left(\frac{4\,\beta\,H}{\sqrt{d_{qk}}}\right)^{\frac{3^L-1}{2}} \|res(\boldsymbol{X})\|^{3^L}_{1,\infty}.$$

The bound guarantees convergence of $\mathrm{SAN}(\boldsymbol{X})$ to rank one when $4\beta H < \sqrt{d_{qk}}$.

The identified cubic rate of convergence is significantly faster than what would be expected when analyzing products of stochastic matrices (linear rate). As a rule of thumb, to achieve a decline of three orders of magnitude, say from 1000 to 1, one could expect a linear rate of convergence to require roughly a dozen iterations, whereas a cubic rate can do so in just two or three iterations. The reason why we get a cubic rate is that the rank of attention matrices depends also on the rank of the input. As we show, the self-attention heads mix tokens faster when formed from a low-rank matrix. This phenomenon becomes stronger as we build deeper SANs, leading to a cascading effect.

1.3 Skip connections counteract rank collapse

Our findings raise a pertinent question—why do attention-based networks work in practice if attention degenerates to a rank-1 matrix doubly exponentially with depth? Aiming to obtain a deeper understanding, we focus on the transformer architecture [4] and expand our analysis by incorporating a key component of transformers that SANs lack: *skip connections*. While we can derive an upper bound for the residual similar to above, it is more informative to have a *lower* bound on the residual, to align with practice, where SANs with skip connections do not suffer rank collapse. We present the following simple lower bound:

Theorem 2. *Consider a depth-L and width-H SAN with skip connections. There exist infinitely many parameterizations for which $res(\boldsymbol{X}^L) \geq res(\boldsymbol{X})$. The preceding holds even for $L \to \infty$ and β arbitrarily small.*

1.4 Rank collapse in practice

To verify our theoretical predictions, we examine the residual of a well-known transformer architecture: BERT [2]. Figure 1 plots the relative residual of each layer's output before and after the network has been trained. To compute these ratios we ran the network on 32 samples of 128 tokens excerpts of biographies from Wikipedia [3] and display the mean and standard deviation.

The experiment confirms that, as soon as the skip connections are removed, all networks exhibit a rapid rank collapse.

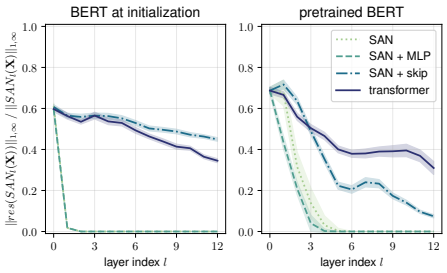


Fig. 1: Relative norm of the residual along the depth before and after training. Pure attention (SAN) converges rapidly to a rank-1 matrix. Adding MLP blocks and skip connection gives a transformer. Skip connections play a critical role in mitigating rank collapse (i.e., a zero residual).

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: International Conference on Learning Representations (2015)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. CoRR (2018), arXiv:1810.04805
3. Lebret, R., Grangier, D., Auli, M.: Generating text from structured data with application to the biography domain. CoRR **abs/1603.07771** (2016), http://arxiv.org/abs/1603.07771
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)
5. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. Advances in Neural Information Processing Systems **32**, 5753–5763 (2019)

Encore Abstract: Interpreting a Black-Box Predictor to Gain Insights into Early Folding Mechanisms*

Isel Grau[1][0000−0002−8035−2887], Ann Nowe[1][0000−0001−6346−4564], and Wim Vranken[1,2,3,4][0000−0001−7470−4324]

[1] Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium
[2] Interuniversity Institute of Bioinformatics in Brussels, Belgium
[3] Structural Biology Brussels, Vrije Universiteit Brussel, Belgium
[4] VIB Structural Biology Research Centre, Belgium

This document is an extended abstract of the paper "Interpreting a black-box predictor to gain insights into early folding mechanisms" published at the Computational and Structural Biotechnology Journal [2].

1 Motivation

Proteins perform a multitude of essential functions in nature. The protein sequence encodes its behavior and, by extension, the environmental context that is required for the protein to fold and/or function. From the different theories about how proteins fold independently, the concept of initial *foldon* formation is now strongly supported by hydrogen-deuterium exchange (HDX) based mass spectrometry (MS) experiments [1, 5]. Foldons essentially form through favorable interactions between amino acids close to each other in the sequence (early folding residues or EFRs), so further creating local structural elements that provide the right context for other residues in the protein to fold.

To gain insights in the early folding residues that drive very first stage of protein folding and the subsequent formation of foldons, the Start2Fold database was created [7]. Based on the Start2Fold per-residue information for a set of 30 proteins, the EFoldMine predictor [8] uses a support vector machine to detect the location of likely early folding residues in a protein sequence. Although support vector machines are known to be highly accurate classifiers based on strong mathematical foundations, the resulting model in multi-dimensional space is difficult, if not impossible, to understand by humans. This restricts the extraction of further knowledge about the determinants of early folding in proteins.

* W.V. is supported by the Research Foundation Flanders (FWO) - project [grant number G.0328.16 N]. I.G. is supported by the Flemish Government (AI Research Program) and the BRIGHTanalysis project, funded by the European Regional Development Fund (ERDF) and the Brussels-Capital Region.

2 Results

To enable interpretation of the EFR determinants, we propose a semi-supervised classification approach, where we leverage unlabeled and non-homologous protein sequence data for which protein structure data are available [9]. By labeling these data with EFR residues as identified by the black-box approach, we enlarge the interpretable training data, assuming it helps in elucidating the separation of the classes by interpretable classifiers. The goal is to obtain an interpretable model with better performance compared to only using experimentally labeled data, as well as obtaining a large dataset of (predicted) early folding data that can be analyzed statistically.

Our self-labeling grey-box (SlGb) approach [3, 4] therefore aims to find a balance between accuracy and interpretability in a semi-supervised classification setting, so leveraging both labeled and unlabeled data, and providing a more flexible approach to interpretability. In the learning process, the enlarged interpretable dataset is amended to avoid propagating misclassifications in the self-labeling. We experiment with rule-based classifiers as a proxy for interpretability, since these approaches are capable of providing both global holistic views of the model and local interpretations that explain a particular prediction.

We show that the self-labeling grey-box approach achieves competitive results against the EFoldMine black box in terms of sensitivity and specificity, through a leave-one-group-out cross-validation. Yet, it is able to represent the classification model with an average of 43 rules. Further analysis of these rules, combined with more classical analyses of the enlarged predicted dataset, enables us to gain mechanistic residue-level insights into the early folding process as well as a better definition of what constitutes an early folding fragment, which can provide useful information for protein design strategies.

The prediction rules are fully interpreted for the SlGb approach, and analyzed in relation to sequence patterns and secondary structure adopted in the folded protein, with all information provided via `http://xefoldmine.bio2byte.be/`, a resource for the community to help understand and steer early protein folding. Our interpretation confirms the importance of backbone rigidity for early folding [6], and reveals the importance of inherent sheet propensity for the early folding residue itself, and strong helix propensity for the residue at position -2. This indicates that very particular specific restrictions on local conformations could be driving the formation of more stable local structures that then initiate the folding process.

References

1. Englander, S.W., Mayne, L.: The nature of protein folding pathways (2014). https://doi.org/10.1073/pnas.1411798111
2. Grau, I., Nowé, A., Vranken, W.: Interpreting a black box predictor to gain insights into early folding mechanisms. Computational and Structural Biotechnology Journal (2021). https://doi.org/10.1016/j.csbj.2021.08.041
3. Grau, I., Sengupta, D., Garcia Lorenzo, M.M., Nowé, A.: Interpretable self-labeling semi-supervised classifier. In: IJCAI/ECAI 2018 Workshop on Explainable Artificial Intelligence (XAI) (2018)
4. Grau, I., Sengupta, D., Garcia Lorenzo, M.M., Nowé, A.: An Interpretable Semi-supervised Classifier using Rough Sets for Amended Self-labeling. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE (2020)
5. Hu, W., Walters, B.T., Kan, Z.Y., Mayne, L., Rosen, L.E., Marqusee, S., Englander, S.W.: Stepwise protein folding at near amino acid resolution by hydrogen exchange and mass spectrometry. Proceedings of the National Academy of Sciences of the United States of America **110**(19), 7684–7689 (2013). https://doi.org/10.1073/pnas.1305887110
6. Pancsa, R., Raimondi, D., Cilia, E., Vranken, W.F.: Early Folding Events, Local Interactions, and Conservation of Protein Backbone Rigidity. Biophysical Journal **110**(3), 572–583 (2016). https://doi.org/10.1016/j.bpj.2015.12.028
7. Pancsa, R., Varadi, M., Tompa, P., Vranken, W.F.: Start2Fold: a database of hydrogen/deuterium exchange data on protein folding and stability. Nucleic acids research **44**(D1), D429–D434 (2016). https://doi.org/10.1093/nar/gkv1185
8. Raimondi, D., Orlando, G., Pancsa, R., Khan, T., Vranken, W.F.: Exploring the Sequence-based Prediction of Folding Initiation Sites in Proteins. Scientific Reports **7**(1), 1–11 (2017). https://doi.org/10.1038/s41598-017-08366-3
9. Wang, G., Dunbrack, R.L.: PISCES: Recent improvements to a PDB sequence culling server. Nucleic Acids Research **33**(SUPPL. 2), W94 (2005). https://doi.org/10.1093/nar/gki402

FOLASP: FO(·) as Input Language for Answer Set Solvers

Kylian Van Dessel, Jo Devriendt, and Joost Vennekens

KU Leuven, Dept. of Computer Science, De Nayer Campus, Sint-Katelijne-Waver, Belgium
Leuven.AI – KU Leuven Institute for AI, Leuven, Belgium

Over the past decades, Answer Set Programming (ASP) has emerged as an important paradigm for declarative problem solving [5]. Technological progress in ASP has been stimulated by the use of common standards, such as the ASP-Core-2 language. While ASP has its roots in non-monotonic reasoning, efforts have also been made to reconcile ASP with classical first-order logic (FO). This has resulted in the development of FO(·) [3], an expressive extension of FO, which allows ASP-like problem solving in a purely classical setting. This language may be more accessible to domain experts already familiar with FO, and may be easier to combine with other formalisms that are based on classical logic. It is supported by the IDP inference system [2], which has successfully competed in a number of ASP competitions. However, technological progress has been hampered by the limited number of systems that are available for FO(·). We address this gap by developing FOLASP: a translation tool that transforms an FO(·) specification into ASP-Core-2, thereby allowing ASP-Core-2 solvers to be used as solvers for FO(·) as well.

An IDP specification consists of three parts, a vocabulary, a structure, and a theory. Using graph coloring as an example, we now illustrate how each of these is translated to ASP.

Vocabulary V {
  $type\ Country$
  $type\ Color$
  $Border(Country, Country)$
  $ColorOf(Country) : Color$
}

$\{colorOf(C, X)\}$:- $country(C), color(X).$
$\delta_r(C)$:- $\#count\{C, X : colorOf(C, X)\} = 1.$
:- $\#count\{C : \delta_r(C)\} \neq 3.$

Structure S: V {
  $Country = \{be, nl, lux\}$
  $Color = \{red, blue\}$
  $Border = \{nl, be; be, lux\}$
}

$country(be).\ country(nl).\ country(lux).$
$color(red).\ color(blue).$
$border(nl, be).\ border(be, lux).$

Theory T: V {
$\forall c_1, c_2 \colon Border(c_1, c_2)$
$\Rightarrow ColorOf(c_1) \neq ColorOf(c_2).$
}

$\delta_{r3}(C_1, C_2, X) \text{ :- } not\ colorOf(C_1, X).$
$\delta_{r3}(C_1, C_2, X) \text{ :- } not\ colorOf(C_2, X).$
$\delta_{r2}(C_1, C_2) \text{ :- } \#count\{X \colon \delta_{r3}(C_1, C_2, X)\} = 2.$
$\delta_{r2}(C_1, C_2) \text{ :- } not\ border(C_1, C_2).$
$\text{ :- } \#count\{C_1, C_2 \colon \delta_{r2}(C_1, C_2)\} \neq 9.$

Note that only vocabulary symbols not closed by their interpretation in the structure are translated and that more complex language features of FO(·), such as function symbols or logical implications, are reduced to a more basic form.

By implementing the translation, we created a new model expansion engine for FO(·), called FOLASP. [1] It uses the syntax of the IDP system for its input and output, and uses Clingo [4] as back-end ASP solver. We tested the correctness of the FOLASP by checking that the solutions produced by FOLASP are accepted as such by IDP, and that, for optimization problems, the optimal objective values produced by IDP and FOLASP were in agreement.

In our experiments, we compare FOLASP using Clingo as backend ASP solver to two other approaches: IDP, running on the same FO(·) specifications as FOLASP, and Clingo, running on native ASP encodings of the same problems. As benchmark set, we use the problem instances from the *model-and-solve* track of the fourth ASP competition [1]. We use fourteen benchmark families that cover a wide range of applications, with each (close to) thirty instances. [2] Our experiments confirmed what one would typically expect, namely that the best performance is obtained by running a specification that was native to a particular solver on that solver. However, the experiments also showed that, for a number of benchmarks, our translation-based approach is actually able to match or even, in rare cases, outperform the native approaches. This demonstrates the usefulness of our translation also from a computational perspective: a specification that performs poorly with one solver, may be more efficient when translated to the input language of another solver. Our experiments also demonstrate that, in cases where the translation performs significantly worse than the native solutions, the grounding often is very large. Future work will therefore focus on further optimising the translation to reduce the overhead it introduces.

In conclusion, by presenting a translation of FO(·) model expansion problems to ASP-Core-2, we both extend the range of solvers for FO(·) and enable the use of FO(·) as an alternative modeling language for these solvers. Hereby, the main contribution of our work is to provide increased flexibility, both in choice of specification language and in choice of solver. We believe that this will be useful to drive technological progress, to develop real-world applications using the best tools for the job, and to allow cross-fertilisation between different research groups.

[1] The source code is published on Gitlab: `https://gitlab.com/EAVISE/folasp`
[2] Runnable software, instance files, and detailed experimental results are made available at Zenodo: `https://doi.org/10.5281/zenodo.4771774`

References

1. Alviano, M., Calimeri, F., Charwat, G., Dao-Tran, M., Dodaro, C., Ianni, G., Krennwallner, T., Kronegger, M., Oetsch, J., Pfandler, A., Pührer, J., Redl, C., Ricca, F., Schneider, P., Schwengerer, M., Spendier, L.K., Wallner, J.P., Xiao, G.: The fourth Answer Set Programming competition: Preliminary report. In: Cabalar, P., Son, T.C. (eds.) Logic Programming and Nonmonotonic Reasoning, 12th International Conference, LPNMR 2013, Corunna, Spain, September 15-19, 2013. Proceedings. LNCS, vol. 8148, pp. 42–53. Springer (2013), http://dx.doi.org/10.1007/978-3-642-40564-8_5
2. Bruynooghe, M., Blockeel, H., Bogaerts, B., De Cat, B., De Pooter, S., Jansen, J., Labarre, A., Ramon, J., Denecker, M., Verwer, S.: Predicate logic as a modeling language: modeling and solving some machine learning and data mining problems with IDP3. TPLP **15**(6), 783–817 (November 2015). https://doi.org/10.1017/S147106841400009X, http://journals.cambridge.org/article_S147106841400009X
3. Denecker, M., Ternovska, E.: A logic of nonmonotone inductive definitions. ACM Trans. Comput. Log. **9**(2), 14:1–14:52 (Apr 2008), http://dx.doi.org/10.1145/1342991.1342998
4. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Multi-shot ASP solving with clingo. TPLP **19**(1), 27–82 (2019). https://doi.org/10.1017/S1471068418000054, https://doi.org/10.1017/S1471068418000054
5. Marek, V., Truszczyński, M.: Stable models and an alternative logic programming paradigm. In: Apt, K.R., Marek, V., Truszczyński, M., Warren, D.S. (eds.) The Logic Programming Paradigm: A 25-Year Perspective, pp. 375–398. Springer-Verlag (1999), http://arxiv.org/abs/cs.LO/9809032

On Explainable Negotiations via Argumentation

Victor Contreras[1][0000−0002−6189−0217], Reyhan Aydoğan[2,3][0000−0002−5260−9999], Amro Najjar[4][0000−0001−7784−6176], and Davide Calvaresi[1][0000−0001−9816−7439]

[1] University of Applied Sciences Western Switzerland, Switzerland
name.surname@hevs.ch
[2] University of Luxembourg, Luxembourg
amro.najjar@uni.lu
[3] Department of Computer Science, Özyegin University, Istanbul, Turkey
reyhan.aydogan@ozyegin.edu.tr
[4] Interactive Intelligence Group, Delft University of Technology, Netherlands

1 Introduction & Background

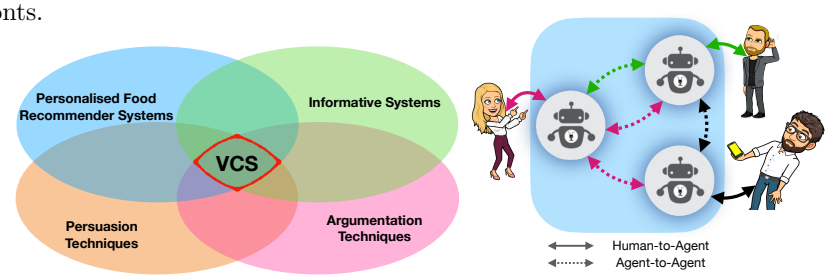
Modern society performs countless choices affecting all sorts of needs daily. Both industry and academia are intensifying their effort to both extend the plethora of possible alternatives and narrow down the most significant ones to be suggested to the user [1]. Thus, it would maximize the possibility of the services consumption and user satisfaction. Recommender systems (RS) [2] have reached remarkable accuracy and efficacy in several domains [3]. Nevertheless, more sensitive areas (i.e., nutrition) demand more complex dynamics beyond conventional RS' capabilities. For example, virtual coaching systems (VCS) leverage persuasion techniques, argumentation, informative systems, and RS (see Figure 1a). However, their efficacy is still far from the one achieved by human coaches, even considering the limitations of the case (see [4]). In particular, the major setbacks are the lack of explanations supporting a given suggestion, the impossibility of "discussing" it with the VCS, and the lack of significant explorations for new out-of-the-box solutions.

Therefore, this work suggests the following negotiation schema for nutrition VCS: $1 - to - 1(-to - \sigma)$ with $\sigma = 0, ..., N$ and N being the number of virtual VCs in the system. In particular, it leverages human-to-agent $(1 - to - 1)$ and agent-to-agent $(1 - to - \sigma)$ joint problem solving via negotiation to generate recommendations and arguments to support them.

2 Personalized Health Coach: Vision & Challenges

Our approach envisions a one-to-one user-agent mapping. Nevertheless, the VCS can consist of multiple agents (assisting users possibly characterized by partially shared traits/features). Therefore, the possibility of extending the agent's knowledge and range of recommendations leveraging other agents' knowledge is more than tangible. Let us assume a user is interacting with the associate agent who has insufficient data to provide accurate suggestions (i.e., cold start). To avoid less appealing and possibly wrong assumptions/suggestions, the agent must profit from inter-agent negotiations to convene more accurate support (see

Figure 1b). With such interactions, the freshman agents produce a series of negotiations equipped with proper argumentations. Once an agreement is reached, new knowledge can be generated, or the old one can be revised. This framework can be formulated as a team negotiation [5]. The team representative (e.g., a freshman agent) can negotiate with the user while, at the same time, it can negotiate with other expert agents. The features used in the agent-to-agent negotiation [6] can exploit or explore solutions leveraging the agents' understanding over personal information (without ever disclosing the actual personal data) and previous interactions. To do so, the first challenges to be overcome are:
CH1 - Effective Interaction: Both structured and natural language-based interactions need to define common ground. Therefore, the challenge is to establish shared syntax, semantic, and knowledge representations. **CH2 - Generating Explainable Arguments:** Comprehensive, personalized, and well-structured explanations can enhance the recommendations' acceptability. The challenge is to create techniques to dynamically generate interpretable explanations (e.g., in natural language or images) w.r.t. their interests and background. **CH3 - Explainable Negotiations:** The interactions must produce sound outcomes (i.e., the decision should be supported by interpretable arguments and suggestions) [7–9]. The challenge is to design agents capable of reasoning over the negotiation, handling information requests, users' demands/interests dynamically, and accordingly generating an offer (i.e., recommendation) equipped with the breakdown of the reasoning process. In addition, the agent should be able to process and learn from users' feedback/comments (e.g., why a given offer is not acceptable). **CH4 - Simultaneous negotiations:** If short on resources/data, agents can help each other sharing their experiences. It can be formulated as group negotiation(s), exchanging aggregated (explainable) understandings on multiple fronts.



(a) VCS Overview. (b) Explainable Team Negotiation

Fig. 1: Vision and Negotiation Framework

Acknowledgments

This work has been partially supported by the CHIST-ERA grant CHIST-ERA-19-XAI-005, and by *(i)* the Swiss National Science Foundation (G.A. 20CH21_195530), *(ii)* the Italian Ministry for Universities and Research, *(iii)* the Luxembourg National Research Fund (G.A. INTER/CHIST/19/14589586 and INTER/Mobility/19/13995684/DLAI/van), *(iv)* the Scientific and Research Council of Turkey (TÜBİTAK, G.A. 120N680).

References

1. Davide Calvaresi, Giovanni Ciatto, Amro Najjar, Reyhan Aydoğan, Leon Van der Torre, Andrea Omicini, and Michael Schumacher. Expectation: Personalized explainable artificial intelligence for decentralized agents with heterogeneous knowledge. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 331–343. Springer, 2021.
2. Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.
3. Mouzhi Ge, Francesco Ricci, and David Massimo. Health-aware food recommender system. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 333–334, 2015.
4. Bart A Kamphorst. E-coaching systems. *Personal and Ubiquitous Computing*, 21(4):625–632, 2017.
5. Victor Sanchez-Anguix, Reyhan Aydoğan, Vicente Julian, and Catholijn M. Jonker. Intra-Team Strategies for Teams Negotiating Against Competitor, Matchers, and Conceders. In Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Takayuki Ito, Minjie Zhang, Quan Bai, and Katsuhide Fujita, editors, *Novel Insights in Agent-based Complex Automated Negotiation*, pages 3–22. Springer Japan, Tokyo, 2014.
6. Shaheen Fatima, Sarit Kraus, and Michael Wooldridge. *Principles of Automated Negotiation*. Cambridge University Press, New York, NY, USA, 1st edition, 2014.
7. Sarit Kraus, Katia Sycara, and Amir Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1):1 – 69, 1998.
8. Iyad Rahwan, Sarvapali D Ramchurn, Nicholas R Jennings, Peter Mcburney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, 2003.
9. Onat Güngör, Umut Çakan, Reyhan Aydoğan, and Pinar Özturk. Effect of awareness of other side's gain on negotiation outcome, emotion, argument, and bidding behavior. In Reyhan Aydoğan, Takayuki Ito, Ahmed Moustafa, Takanobu Otsuka, and Minjie Zhang, editors, *Recent Advances in Agent-based Negotiation*, pages 3–20, Singapore, 2021. Springer Singapore.

Expert RuleFit: Complementing Rule Ensembles with Expert Knowledge

Luisa Ebner[1], Malte Nalenz[2], Annette ten Teije[2], Frank van Harmelen[1], and Thomas Augustin[2]

[1] Vrije Universiteit Amsterdam, de Boelelaan 1081a, Amsterdam, NL
[2] University of Munich, Ludwigstr. 33, Munich, Germany
Annette.ten.Teije}@vu.nl

Abstract. We present Expert RuleFit (ERF), an approach to integrate expert knowledge in the form of rules and linear terms into an existing method for rule learning (RuleFit). A customized regularization strategy allows us to consider the different strengths of expert knowledge. For an empirical evaluation, we trained ERF models on a diabetes dataset for which we acquired expert rules from medical guidelines and expert interviews. The integration of different knowledge sources makes the ERF model a promising tool for learning accurate, explainable and trustworthy medical decision rules.

Machine Learning (ML) systems offer great potential to provide healthcare improvements. However, they often generalise poorly on small training sets, are difficult to combine with human expertise, and are often difficult to explain to experts. We hypothesise that the inclusion of prior expert knowledge will allow ML algorithms to better generalize to unseen cases while allowing human experts to better understand and validate recommendations. We test this hypothesis by proposing Expert RuleFit (ERF), a classification method that combines the strengths of inductive ML with expert rule-based reasoning. ERF injects expert knowledge in the form of rules and linear terms into the existing rule ensemble method RuleFit [2]. A tailored regularization strategy allows experts to specify *confirmed* knowledge to be certainly included into the final prediction model as well as *optional* knowledge to be soft-promoted over data rules through a customized penalization strategy.

METHOD Our proposed method *Expert* RuleFit (ERF) operates in 3 stages: **Stage 1: Knowledge Acquisition.** Prior to the learning process, expert knowledge is formulated as rules and linear effects. Useful sources for rule formulation are clinical practice guidelines, whose recommendations are often formulated as rule-like statements. To distinguish different degrees of validated expert knowledge, ERF allows rules and linear terms to be declared as *confirmed* or *optional*. **Stage 2: Combined Ensemble Generation.** Consequently, 4 sets of expert

[0] Full paper published in the 12th International Workshop on Knowledge Representation for Healthcare (KR4HC), LNCS, Springer Verlag, forthcoming.

knowledge enter the ERF model together with the given dataset: confirmed expert rules and linear terms, as well as their optional counterparts.
Stage 3: Knowledge-Aware Regularization. To learn the coefficients for all of these elements, we developed a tailored regularization strategy, where adaptive *penalty factors* serve to guarantee the inclusion of confirmed expert knowledge and allow for a soft-promotion of optional expert knowledge over data-generated predictors in the final model.

EXPERIMENTS We evaluated ERF on the PID dataset of 768 Diabetes Type 2 patients from the UCI repository [1]. The task is to diagnose these patients based on 10 observable values. As expert knowledge, we manually extracted rules and linear terms from two diabetes guidelines. In two expert interviews, practicing physicians specified task-relevant patient subpopulations based on their diagnostic expertise. This resulted in 20 confirmed expert rules, 2 confirmed linear terms, 34 optional expert rules and 3 optional linear terms.
Experimental Protocol. We trained four different versions of our proposed ERF model, plus an existing implementation of the conventional RuleFit model and a Random Forest model (the latter two serve as baselines). The four versions of the ERF model differ in the extent to which they penalise data rules over expert knowledge. We train the model on successively smaller subsets of the data. We applied 10-fold cross validation to provide balanced accuracy measures.
Results. AUC and classification accuracy results are similar for all model settings and data-set sizes. This shows that expert knowledge is often able to replace data-driven rules without sacrificing predictive performance. For smaller samples, the expert knowledge contains as much task-relevant information as 400 training examples. We found that the inclusion of expert knowledge decreases the ensemble size compared to our baseline implementation of RuleFit: 50-75% of all base learners that remain in the final model and 8-10 out of the 10 most important terms (i.e. the terms with highest coefficients) correspond to expert knowledge. Thus, ERF models largely base their results on validated, medically coherent predictors instead of correlations derived from a patient sample. These results were confirmed in a simulation study, as well as in a Diabetes Type 2 hospital readmission prediction task on 100.000 patients.

CONCLUSION We conclude that the ERF replaces data-driven rules with explainable and medically coherent rules without sacrificing predictive accuracy or adding to model complexity, while needing fewer training data. As such, ERF promises accurate and yet simple models, including both data-driven rules and a large fraction of validated and explainable expert-provided knowledge.

[1] Dua, D., Graff, C.: UCI machine learning repository, http://archive.ics.uci.edu/ml2
[2] Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. *The Annals of Applied Statistics* **2**(3), 916–954 (2008)

Active Monitoring of Neural Networks

Anna Lukina[1][0000−0001−9525−0333], Christian Schilling[2][0000−0003−3658−1065], and Thomas A. Henzinger[3][0000−0002−2985−7724]

[1] Delft University of Technology, Delft, The Netherlands
a.lukina@tudelft.nl
[2] University of Konstanz, Konstanz, Germany
christian.schilling@uni-konstanz.de
[3] IST Austria, Klosterneuburg, Austria
thomas.henzinger@ist.ac.at

Abstract. Neural-network classifiers are trained to achieve high prediction accuracy. However, their performance still suffers from frequently appearing inputs of unknown classes. As a component of a cyber-physical system, the classifier in this case can no longer be reliable and is typically retrained. We propose an algorithmic framework for monitoring reliability of a neural network. In contrast to static detection, a monitor wrapped in our framework operates in parallel with the classifier, communicates interpretable labeling queries to the human user, and incrementally adapts to their feedback.

Keywords: monitoring · neural networks · novelty detection.

Automated classification is an essential part of numerous modern technologies and one of the most popular applications of deep neural networks [4]. Neural-network image classifiers have fast-forwarded technological development in many research areas, e.g., automated object localization as a stepping stone to successful real-world robotic applications [9]. Such applications require a high level of reliability from the neural networks.

However, when deployed in the real world, neural networks face a common problem of novel input classes appearing at prediction time, leading to possible misclassifications and system failures. For example, consider a scenario of a neural network used for labeling inputs and making decisions about the next actions for an automated system with limited human supervision: a robot assistant learning to recognize objects in a new home. Assume the neural network is trained well on a dataset containing examples of a finite set of classes. However, after this robot is deployed in the real home, novel classes of objects can appear and confuse the neural network. The inherent misclassifications can stay undetected and accumulate over time, eventually reducing overall accuracy.

The likelihood of severe system damage increases with the frequency and diversity of novel input classes. Typically, this risk is addressed by detecting novel inputs, augmenting the training dataset, and retraining the classifier from scratch [5]. This procedure is not only inefficient, but also leaves the system

vulnerable until such a dataset has been collected. Techniques to incrementally adapt classifiers at prediction time are beneficial for improving accuracy in real-world applications [8, 7]. They, however, do not provide desired interpretability for the human. Approaches to run-time monitoring of neural networks were therefore introduced [6]. In particular, approaches based on abstractions [2, 3, 1, 10] proved to be effective at detecting novel input classes. In addition, they provide transparency of neural-network monitoring.

Crucially, these monitors are constructed offline and remain static at prediction time. Functionalities they are still lacking are distinguishing between "known" and "unknown" novelties and selectively adapting at prediction time.

We propose an active monitoring framework for neural networks that detects novel input classes, obtains the correct labels from a human authority, and adapts the neural network and the monitor to the novel classes, all at prediction time. The framework contains a mechanism for automatic switching between monitoring and adaptation based on run-time statistics. Adaptation consists of either learning new classes (when enough data has been collected) or retraining with more up-to-date information (when the run-time performance is unsatisfactory), where retraining is applied to the network and the monitor independently. A trained neural-network model accompanied by our framework, as an external observer and mediator between the neural network and the human, achieves improved transparency of operation through informative interaction.

Furthermore, we propose a new monitor designed for the adaptive setting. Introducing a quantitative metric at the hidden layers of the neural network, the monitor timely warns about inputs of novel classes and reports its own confidence to the authority. This allows for assessing the need of model adaptation. The quantitative metric allows for easy adaptation at prediction time to newly introduced labels and successfully maintains overall classification accuracy on inputs of known and previously novel classes combined. As such, our framework is an interactive and interpretable tool for informed decision making in neural-network based applications.

Our framework is independent of the choice of the dataset and the neural-network architecture. The only requirements for applicability of our approach are access to the output of the feature layer(s). We plan to extend our procedure toward real-world applications with particular need of active monitoring, e.g., in robotics for the trained controller to gradually adapt to the behavior of the authority. Other interesting directions are time-critical applications where the adaptation of the monitor or the neural network need to be delayed to uncritical phases, and scenarios where novel inputs occur rarely. In addition, the underlying method of our framework can serve as a suitable tool for designing an algorithmic approach to explainability of a neural network's predictions.

References

1. Chen, Y., Cheng, C., Yan, J., Yan, R.: Monitoring object detection abnormalities via data-label and post-algorithm abstractions. CoRR **abs/2103.15456** (2021), https://arxiv.org/abs/2103.15456

2. Cheng, C., Nührenberg, G., Yasuoka, H.: Runtime monitoring neuron activation patterns. In: DATE. pp. 300–303. IEEE (2019), https://doi.org/10.23919/DATE.2019.8714971
3. Henzinger, T.A., Lukina, A., Schilling, C.: Outside the box: Abstraction-based monitoring of neural networks. In: ECAI. Frontiers in Artificial Intelligence and Applications, vol. 325, p. 2433–2440. IOS Press (2020), http://doi.org/10.3233/FAIA200375
4. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. Neurocomputing **234**, 11–26 (2017), https://doi.org/10.1016/j.neucom.2016.12.038
5. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks **113**, 54–71 (2019), https://doi.org/10.1016/j.neunet.2019.01.012
6. Rahman, Q.M., Corke, P., Dayoub, F.: Run-time monitoring of machine learning for robotic perception: A survey of emerging trends. IEEE Access **9**, 20067–20075 (2021), https://doi.org/10.1109/ACCESS.2021.3055015
7. Rebuffi, S., Kolesnikov, A., Sperl, G., Lampert, C.H.: iCaRL: Incremental classifier and representation learning. In: CVPR. pp. 5533–5542. IEEE Computer Society (2017), https://doi.org/10.1109/CVPR.2017.587
8. Royer, A., Lampert, C.H.: Classifier adaptation at prediction time. In: CVPR. pp. 1401–1409. IEEE Computer Society (2015), https://doi.org/10.1109/CVPR.2015.7298746
9. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: IROS. pp. 23–30. IEEE (2017), https://doi.org/10.1109/IROS.2017.8202133
10. Wu, C., Falcone, Y., Bensalem, S.: Customizable reference runtime monitoring of neural networks using resolution boxes. CoRR **abs/2104.14435** (2021), https://arxiv.org/abs/2104.14435

A Gaze-Based Measure of Temporal Salience

V. Javier Traver[1], Judith Zorío[1], and Luis A. Leiva[2]

[1] Institute of New Imaging Technologies, Universitat Jaume I, Spain
[2] ILIAS, University of Luxembourg, Luxembourg
vtraver@uji.es, al258412@alumail.uji.es, name.surname@uni.lu

Abstract. Temporal salience considers how visual attention varies over time. Although visual salience has been widely studied from a spatial perspective, its temporal dimension has been mostly ignored, despite arguably being of utmost importance to understand the temporal evolution of attention on dynamic contents. To address this gap, we proposed GLIMPSE, a novel measure to compute temporal salience based on the observer-spatio-temporal consistency of raw gaze data. The measure is conceptually simple, training free, and provides a semantically meaningful quantification of visual attention over time. GLIMPSE could serve as the basis for several downstream tasks such as segmentation or summarization of videos. Our software and data are publicly available.

Keywords: visual attention · temporal salience · eye-gaze · video

1 Introduction and Method Overview

How to automatically estimate temporal salience in videos using eye-tracking data? Spatial saliency is well understood [1,2,3,4,5,6,7,9,10,11,12,13,14,16,18,19], however the temporal dimension has been mostly ignored [8,15,20]. Our main hypothesis was that when gaze coordinates are spatio-temporally consistent across multiple observers, it is a strong indication of visual attention being allocated at a particular location within a frame (spatial consistency) and at a particular time span (temporal consistency).

Our approach, named GLIMPSE [17] (gaze's spatio-temporal consistency from multiple observers), is illustrated in Figure 1 and formulated as follows:

$$s(t) = \frac{2}{n(n-1)} \sum_{\substack{i,j \in \{1,\ldots,n\} \\ i \neq j}} \mathbb{1}[d_{ij} < \theta_s], \quad t \in \{1, \ldots, T\} \tag{1}$$

where d_{ij} is the pairwise Euclidean distance between the ith and jth points in the set $\mathcal{P}_t$ of n gaze points from all the observers within a temporal window of length $2\theta_t + 1$ centered at frame t, i.e.,

$$\mathcal{P}_t = \Big\{ \mathbf{g}(o,t) : o \in \{1, \ldots, N\},\ t \in [t - \theta_t, t + \theta_t] \Big\} \tag{2}$$

and $\mathbb{1}[p]$ is the indicator function, which is 1 when predicate p is true and 0 otherwise.

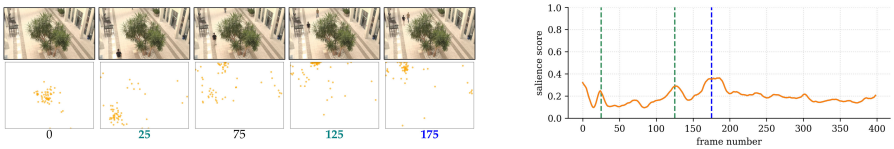


Fig. 1. Left: Frames of SAVAM video v22 (top) and observers' gaze points (bottom). Right: temporal salience score estimation with pointers to key events.

GLIMPSE processes raw gaze data and has two hyperparameters (θ_t and θ_s) to control the temporal and spatial scale. The effect of such hyperparameters is shown in Figure 2. Note that $s(t) \in [0, 1]$, where 1 means high temporal salience. Finally, Figure 3 shows a convergence analysis to decide how many observers would be required to get good salience estimates. We can see that GLIMPSE is quite scalable, as reliable estimates can be obtained with few observers.

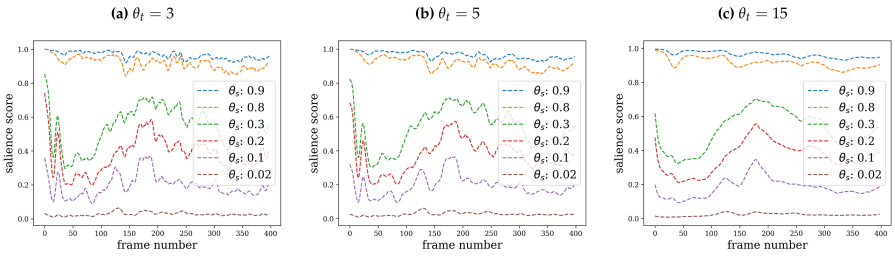


Fig. 2. Effect of spatial scale θ_s and temporal scale θ_t on salience score $s(t)$ computed for SAVAM video v22 (see Figure 1 for an example of the video contents).

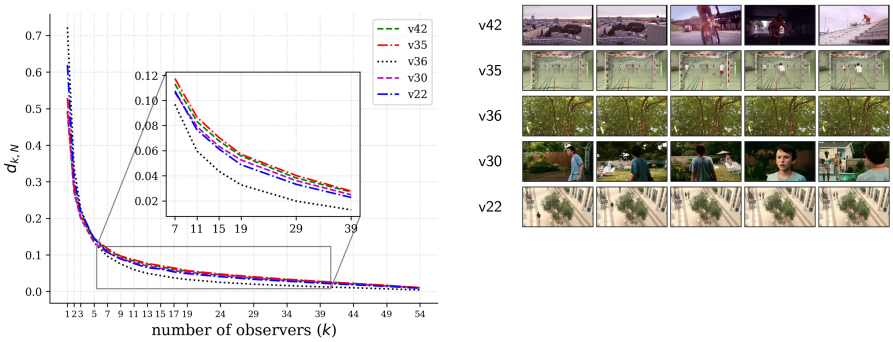


Fig. 3. Convergence analysis for five different SAVAM videos. See [17] for more details.

References

1. Droste, R., Jiao, J., Noble, J.A.: Unified image and video saliency modeling. In: Proc. ECCV (2020)
2. Fosco, C., Newman, A., Sukhum, P., Zhang, Y.B., Zhao, N., Oliva, A., Bylinskii, Z.: How much time do you have? modeling multi-duration saliency. In: Proc. CVPR (2020)
3. Hadizadeh, H., Enriquez, M.J., Bajic, I.V.: Eye-tracking database for a set of standard video sequences. IEEE Trans. Image Process. **21**(2) (2012)
4. Karessli, N., Akata, Z., Schiele, B., Bulling, A.: Gaze embeddings for zero-shot image classification. In: Proc. CVPR (2017)
5. Karthikeyan, S., Thuyen Ngo, Eckstein, M., Manjunath, B.S.: Eye tracking assisted extraction of attentionally important objects from videos. In: Proc. CVPR (2015)
6. Kasprowski, P., Harezlak, K.: Fusion of eye movement and mouse dynamics for reliable behavioral biometrics. Pattern Anal. Appl. **21**(1) (2018)
7. Madsen, J., Júlio, S.U., Gucik, P.J., Steinberg, R., Parra, L.C.: Synchronized eye movements predict test scores in online video education. PNAS **118**(5) (2021)
8. Mahasseni, B., Lam, M., Todorovic, S.: Unsupervised video summarization with adversarial LSTM networks. In: Proc. CVPR (2017)
9. Masciocchi, C.M., Still, J.D.: Alternatives to eye tracking for predicting stimulus-driven attentional selection within interfaces. Hum.-Comput. Interact. **28**(5) (2013)
10. Neves, A.C., Silva, M.M., Campos, M.F.M., do Nascimento, E.R.: A gaze driven fast-forward method for first-person videos. In: Proc. EPIC@ECCV Workshop (2020)
11. Nguyen, T.V., Xu, M., Gao, G., Kankanhalli, M., Tian, Q., Yan, S.: Static saliency vs. dynamic saliency: A comparative study. In: Proc. MULTIMEDIA (2013)
12. Palmero Cantarino, C., Komogortsev, O.V., Talathi, S.S.: Benefits of temporal information for appearance-based gaze estimation. In: Proc. ETRA (2020)
13. Polatsek, P., Benesova, W., Paletta, L., Perko, R.: Novelty-based spatiotemporal saliency detection for prediction of gaze in egocentric video. IEEE Signal Processing Lett. **23**(3) (2016)
14. Salehin, M.M., Paul, M.: A novel framework for video summarization based on smooth pursuit information from eye tracker data. In: Proc. IEEE Intl. Conf. on Multimedia & Expo Workshops (ICMEW) (2017)
15. Sidorov, O., Pedersen, M., Shekhar, S., Kim, N.W.: Are all the frames equally important? In: Proc. CHI EA (2020)
16. Tangemann, M., Kümmerer, M., Wallis, T.S., Bethge, M.: Measuring the importance of temporal features in video saliency. In: Proc. ECCV (2020)
17. Traver, V.J., Zorío, J., Leiva, L.A.: Glimpse: A gaze-based measure of temporal salience. Sensors **21**(9) (2021)
18. Xu, J., Mukherjee, L., Li, Y., Warner, J., Rehg, J.M., Singh, V.: Gaze-enabled egocentric video summarization via constrained submodular maximization. In: Proc. CVPR (2015)
19. Yun, K., Peng, Y., Samaras, D., Zelinsky, G.J., Berg, T.L.: Studying relationships between human gaze, description, and computer vision. In: Proc. CVPR (2013)
20. Zhou, K., Qiao, Y., Xiang, T.: Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In: Proc. AAAI (2018)

Optimizing Reserve Price using Deep Reinforcement Learning and Shaped Reward

Reza Refaei Afshar, Jason Rhuggenaath, Yingqian Zhang, and Uzay Kaymak

Eindhoven University of Technology, Eindhoven, Netherlands

1 Introduction

In [1], we introduced a novel reward shaping method for Deep Reinforcement Learning (DRL) modeling of pricing problems in Real Time Bidding (RTB) auctions. In RTB auctions, the *impressions* generated by user views are the assets to be sold. For each impression, an *ad request* is sent to the intermediate entities between publishers and advertisers. These intermediate entities run online auctions to sell the impressions. The auctions are mainly second price auctions where the winner pays as much as the maximum of the second highest bid and a *reserve price* which determines the minimum price of the impression. In a practical framework used in business, the ad publisher sends ad requests to the Header Bidding Partners (HBPs) simultaneously and receives their bids. Then, a reserve price is set according to the HBPs bids, and another ad request is sent to an AdX. If AdX's winner bid is higher than the highest bid of HBPs, the impression goes to AdX; otherwise, it goes to the HBP. In common practice, the highest bid of HBPs is the reserve price which may not be optimal because AdX might outbid higher reserve prices. Therefore, the problem is to determine the reserve price in impression level to uplift the revenue of ad publishers. Dynamic environment, sequential decision-making modeling of the pricing problem, and limited available information motivate us to use Deep Reinforcement Learning (DRL) to solve this problem. To learn a suitable reward function, our proposed method employs a reward shaping method to prioritize higher reserve prices. Results show that this method increases the revenue significantly.

2 DRL for Reserve Price Optimization

The process of adjusting the reserve price for each impression is performed by using a deep neural network policy that is trained by DRL. We use Proximal Policy Optimization (PPO) algorithm as a policy gradient method to train the policy and the value networks. Common impression information in RTB systems based on HBPs and AdX, including URL, size and location of the ad slot, time of generating the impression, and the highest bid of HBPs (ζ_t^{HBP}) construct the states. The action a_t is the reserve price that is obtained from the output of the policy network. Since a_t is continuous, the policy network has a single output that provides the mean value for a Gaussian distribution. The standard deviation is fixed, and the reserve price a_t is sampled from the Gaussian distribution.

Defining reward is challenging because AdX provides no information about its winner bid (ζ_t^{AdX}) and a binary value (β_t) showing whether the auction has a winner is the only feedback from AdX auction. For this reason, we develop a reward shaping approach that extends the limited responses of the environment. Following the fact that larger a_t may lead to higher revenue, the main objective of the reward shaping is to assign proper weight to larger a_t. To achieve that, the interval between ζ_t^{HBP} and an estimation of ζ_t^{AdX}, is divided into n equal sub-intervals and a particular weight w_j for $j \in \{1, ..., n\}$ is assigned to the interval j. Vector $\vec{w} \in \mathcal{W}$ contains the weights w_j and $\mathcal{W}$ is the space of candidate weights vectors. Vector $\vec{r}(a_t, \zeta_t^{HBP}, \beta_t)$ assigns a reward $r_{t,j}$ to each interval j. Assuming interval zero for values smaller than ζ_t^{HBP} and interval $n+1$ for values larger than the estimated ζ_t^{AdX}, the inner product of vectors $\vec{w} = (w_0, w_1, ..., w_{n+1})$ and $\vec{r}(a_t, \zeta^{HBP}, \beta_t) = (r_{t,0}, r_{t,1}, ..., r_{t,n+1})$ provides the reward value for each impression. In other words, $R(a_t, \zeta^{HBP}, \beta_t)$ is $\vec{r}(a_t, \zeta^{HBP}, \beta_t) \cdot \vec{w}$. The weights and the reward definition are found by searching in the space of candidate values.

3 Experiments and Results

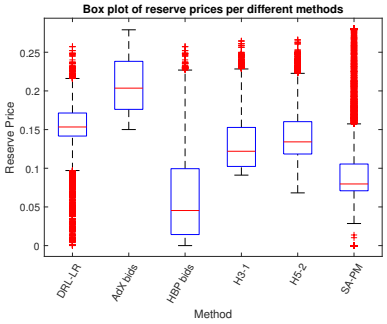


Fig. 1. Revenue of individual impressions

Algorithm	$\sum_t \zeta_t^{HBP}$	$\sum_t a_t$	$\sum_t \zeta_t^{AdX}$	$\%_{a_t}$
DRL-LR		**1527.23**		**73.57%**
H3-1		1315.82		63.38%
H5-3	628.00	1430.96	2075.86	68.93%
DRL-DA		1315.82		63.38%
DRL-RTB		1437.20		69.23%
SA-PM		986.88		47.54%
DRL-LR		**1521.77**		**73.67%**
H3-1		1305.82		63.22%
H5-3	599.32	1420.92	2065.43	68.79%
DRL-DA		1305.82		63.22%
DRL-RTB		1414.03		68.46%
SA-PM		955.09		46.24%

Table 1. Revenue of the algorithms

Since using real RTB systems for training is not possible, we develop a simulator using historical data. For this purpose, we use ζ_t^{HBP} for the impressions that go to AdX, to generate a lower bound for ζ_t^{AdX}. Our proposed method with learned reward is denoted by DRL-LR. Each test data contains 10000 impressions where $\zeta_t^{AdX} > \zeta_t^{HBP}$ because for other impressions the winner is HBP regardless of the value of a_t. As the baselines, we use two heuristic methods (H5-3 and H3-1), DRL with normal feedback of RTB environment as the reward (DRL-RTB), DRL with discrete actions (DRL-DA), our reward shaping DRL method (DRL-LR) and a supervised learning method in the literature (SA-PM). For evaluation, we consider the sum of reserve prices ($\sum_t a_t$), sum of AdX winner bids ($\sum_t \zeta_t^{AdX}$) and their ratio. According to the results shown in table 3 and Fig. 3, DRL-LR outperforms the other pricing methods in terms of revenue.

References

1. Afshar, Reza Refaei, Jason Rhuggenaath, Yingqian Zhang, and Uzay Kaymak. "A Reward Shaping Approach for Reserve Price Optimization using Deep Reinforcement Learning." In The International Joint Conference on Neural Networks (IJCNN2021). 2021.

A Human-Agent Architecture for Explanation Formulation (An extended abstract)*

Yazan Mualla[1], Igor Tchappi[1], Timotheus Kampik[2], Amro Najjar[3], Davide Calvaresi[4], Abdeljalil Abbas-Turki[1], Stéphane Galland[1], and Christophe Nicolle[5]

[1] CIAD, Univ. Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France
[2] Department of Computing Science, Umeå University, 90187 Umeå, Sweden
[3] AI-Robolab/ICR, University of Luxembourg, 4365 Esch-sur-Alzette, Luxembourg
[4] University of Applied Sciences and Arts of Western Switzerland, Sierre, Switzerland
[5] CIAD UMR 7533, Univ. Bourgogne Franche-Comté, UB, F-21000 Dijon, France

1 Introduction

With the widespread use of AI systems, understanding the behavior of intelligent agents and robots is crucial to facilitate successful human-computer interaction (HCI) [3]. Recent studies have confirmed that explaining an agent's behavior to humans fosters the latter's acceptance of the agent [2, 4]. However, providing overwhelming or unnecessary information may also confuse humans and cause failure [15]. For these reasons, *parsimony* has been outlined as one of the key features of successful explanations in HCI [10, 9]; in this context, a parsimonious explanation is defined as the simplest explanation (*i.e.,* least complex) that describes the situation adequately (*i.e.,* descriptive adequacy) [9, 5]. While parsimony is receiving growing attention in the literature, most works are carried out on the conceptual front, and little research has been done from engineering and empirical HCI perspectives.

2 Contribution

This work proposes a mechanism for parsimonious eXplainable AI (XAI) [6, 7, 16]. In particular, it introduces the process of *explanation formulation* and proposes HAExA, a human-agent explainability architecture (Figure 1) allowing to make this formulation operational for remote robots. In HAExA, **remote robots** (right) are represented as agents that generate contrastive explanations[6] [12] to explain their behaviors based on the changes in the environment and their goals. **Assistant agents** (center) collect the remote agents' raw explanations to communicate filtered explanations to the **human** (left); the filtering helps prevent that humans get overwhelmed by the information the remote agents provide. Considering that the assistant agents have a global overview of the environment, they may post-process the raw explanations received from the remote agents to aggregate, update, and filter them; subsequently, they communicate the updated and filtered explanations to the human.

* This work has been accepted in the Journal of Artificial Intelligence on the 2nd of August 2021 [14]. DOI: https://doi.org/10.1016/j.artint.2021.103573

[6] Broadly speaking, contrastive explanations answer *why A and not B?* questions.

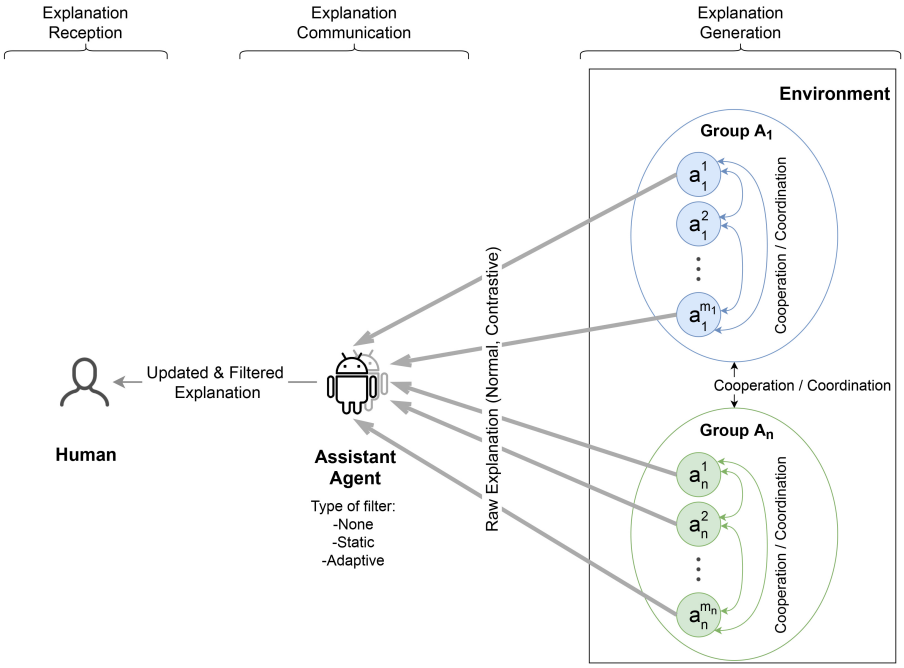


Fig. 1. Human-Agent Explainability Architecture (HAExA).

3 Evaluation and Results

To evaluate HAExA, several research hypotheses are investigated in an HCI study using an agent-based simulation based on a scenario of package delivery in smart cities (*see* our demo paper [13]). The study relies on well-established XAI metrics [8] to estimate how understandable the explanations are to the human participants. The study investigates the impact of the different techniques of explanation formulation (static filter, adaptive filter, and adaptive filter with contrastive explanations) on humans. The participants' responses are collected using a 5-Likert scale [1]. The significance of these responses is statistically analyzed and presented using statistical testing: Non-parametric (Kruskal-Wallis), Parametric (ANOVA), and Cronbach's alpha.

Based on the analysis of *subjective* and *objective understandability*, we gathered evidence that adaptively filtered and contrastive explanations improve human understandability compared to statically filtered explanations (*i.e.,* non-adaptive to the environment). Our insights indicate that contrastive explanations can be used without risking a detrimental effect on understandability. Our study could not confirm the same effect on *trust* (which remains a challenge identified in many other works in the literature [11, 8]). Nevertheless, the results provide empirical insights on human-multiagent system explainability as a starting point that future research on XAI could expand.

References

1. Albaum, G.: The likert scale revisited. Market Research Society. Journal. **39**(2), 1–21 (1997)
2. Anjomshoae, S., Najjar, A., Calvaresi, D., Främling, K.: Explainable agents and robots: Results from a systematic literature review. In: Proc. of 18th Int. Conf. on Autonomous Agents and MultiAgent Systems. pp. 1078–1088. Int. Foundation for Autonomous Agents and Multiagent Systems (2019)
3. Bainbridge, W.A., Hart, J., Kim, E.S., Scassellati, B.: The effect of presence on human-robot interaction. In: RO-MAN 17th IEEE Int. Symposium on Robot and Human Interactive Communication. pp. 701–706 (2008)
4. Calvaresi, D., Mualla, Y., Najjar, A., Galland, S., Schumacher, M.: Explainable multi-agent systems through blockchain technology. In: Calvaresi, D., Najjar, A., Schumacher, M., Främling, K. (eds.) Explainable, Transparent Autonomous Agents and Multi-Agent Systems. pp. 41–58. Springer International Publishing, Cham (2019)
5. Contreras, H.: Simplicity, descriptive adequacy, and binary features. Language pp. 1–8 (1969)
6. Gunning, D.: Explainable artificial intelligence (XAI). Defense Advanced Research Projects Agency (DARPA), nd Web (2017)
7. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: XAI—Explainable Artificial Intelligence. Science Robotics (2019)
8. Hoffman, R.R., Mueller, S.T., Klein, G., Litman, J.: Metrics for explainable AI: Challenges and prospects. arXiv preprint arXiv:1812.04608 (2018)
9. Krizek, G.C.: Ockham's razor and the interpretations of quantum mechanics (2017)
10. Laird, J.: The law of parsimony. The Monist **29**(3), 321–344 (1919), http://www.jstor.org/stable/27900747
11. Madumal, P., Miller, T., Sonenberg, L., Vetere, F.: Explainable reinforcement learning through a causal lens. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 2493–2500 (2020)
12. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence **267**, 1–38 (2019)
13. Mualla, Y., Kampik, T., Tchappi, I.H., Najjar, A., Galland, S., Nicolle, C.: Explainable agents as static web pages: Uav simulation example. In: Calvaresi, D., Najjar, A., Winikoff, M., Främling, K. (eds.) Explainable, Transparent Autonomous Agents and Multi-Agent Systems. pp. 149–154. Springer International Publishing, Cham (2020)
14. Mualla, Y., Tchappi, I., Kampik, T., Najjar, A., Calvaresi, D., Abbas-Turki, A., Galland, S., Nicolle, C.: The quest of parsimonious xai: a human-agent architecture for explanation formulation. Artificial Intelligence p. 103573 (2021)
15. Mualla., Y., Tchappi., I., Najjar., A., Kampik., T., Galland., S., Nicolle., C.: Human-agent explainability: An experimental case study on the filtering of explanations. In: Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 1: HAMT,. pp. 378–385. INSTICC, SciTePress (2020). https://doi.org/10.5220/0009382903780385
16. Ras, G., van Gerven, M., Haselager, P.: Explanation methods in deep learning: Users, values, concerns and challenges. In: Explainable and Interpretable Models in Computer Vision and Machine Learning, pp. 19–36. Springer (2018)

Explainable AI using MAP-independence

Johan Kwisthout[1][0000−0003−4383−7786]

Donders Institute for Brain, Cognition, and Behaviour, Radboud University, Nijmegen, The Netherlands j.kwisthout@donders.ru.nl
http://www.socsci.ru.nl/johank/

Abstract. This is an extended abstract of [5] where we introduce the notion of MAP-independence in Bayesian networks and explore some computational properties of establishing MAP-independence.

Keywords: Bayesian Networks · Most Probable Explanations · Relevance · Explainable AI · Computational Complexity.

1 Motivation

In decision support systems the motivation and justification of the system's diagnosis or classification is crucial for the acceptance of the system by the human user. In Bayesian networks a diagnosis or classification is typically formalized as the computation of the most probable joint value assignment to the hypothesis variables, given the observed values of the evidence variables (known as the MAP problem). While solving MAP gives the most probable explanation of the evidence, the computation is a black box as far as the human user is concerned and it does not give additional insights that allow the user to appreciate and accept the decision. In this paper we specifically try to improve the user's understanding of a specific decision by *explicating the relevant information* that contributed to said decision. In deciding what the best explanation is for a set of observations, marginalizing out unobserved non-hypothesis variables makes the process more opaque: some of these variables have a bigger impact (i.e., are more relevant) on the eventual decision than others, and this information is lost in the process. For example, the absence of a specific test result (i.e., a variable we marginalize out in the MAP computation) may lead to a different explanation of the available evidence compared to when a negative (or positive) test result *were* present. In this situation, this variable is more relevant to the eventual explanation than if the best explanation would be the same, irrelevant of whether the test result was positive, negative, or missing. Our approach in this paper is to motivate a decision by showing which of these variables were relevant in this sense towards arriving at this decision. To this end, we introduce a new concept, MAP-independence, which tries to formally capture this notion of relevance, and explore its role towards a justification of an inference to the best explanation.

2 MAP-independence

Pearl has suggested to use conditional independence as a measure of relevance for explanatory purposes; i.e., a variable is irrelevant when it is conditionally independent of the explanation [7]. We argue that this may be too strict and leaves too many potentially relevant variables. We propose that an explanation is *MAP-independent* from a variable if the explanation will be the same irrespective of any specific value of this variable. Formally, we say that A is MAP-independent from B given $C = c$ when $\forall_{b \in \Omega(B)} \text{argmax}_a \Pr(A = a, B = b \mid C = c) = a$ for a specific value assignment $a \in \Omega(A)$. In decision support systems, an explication of how a variable may impact or fail to impact the most probable explanation of the evidence will both help motivate the system's advice as well as offer guidance in further decisions (e.g., to gather additional evidence [2, 1] to make the MAP explanation more robust).

3 Formal analysis

In the paper we show co-$\mathsf{NP}^{\mathsf{PP}}$-completeness of a suitable decision variant of establishing MAP-independence. A straightforward brute-force algorithm below gives a run-time of $\mathcal{O}(\Omega(\mathbf{R})) = \mathcal{O}(2^{|\mathbf{R}|})$ times the time needed for each MAP computation. This implies that, given known results on fixed-parameter tractability [3] and efficient approximation[6, 4] of MAP, the size of the set against which which we want to establish MAP independence is the crucial source of complexity if MAP can be computed or approximated feasibly.

References

1. van der Gaag, L., Bodlaender, H.: On stopping evidence gathering for diagnostic Bayesian networks. In: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty. pp. 170–181 (2011)
2. van der Gaag, L., Wessels, M.: Selective evidence gathering for diagnostic belief networks. AISB Quarterly **86**, 23–34 (1993)
3. Kwisthout, J.: Most Probable Explanations in Bayesian networks: Complexity and tractability. International Journal of Approximate Reasoning **52**(9) (2011)
4. Kwisthout, J.: Tree-width and the computational complexity of MAP approximations in Bayesian networks. Journal of Artificial Intelligence Research **53**, 699–720 (2015)
5. Kwisthout, J.: Explainable AI using MAP-independence. In: Vejnarová, J., Wilson, N. (eds.) Proceedings of the Sixteenth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty. LNAI, vol. 12897, pp. 1–12. Springer-Verlag (2021)
6. Park, J.D., Darwiche, A.: Complexity results and approximation settings for MAP explanations. Journal of Artificial Intelligence Research **21**, 101–133 (2004)
7. Pearl, J., Paz, A.: GRAPHOIDS: a graph-based logic for reasoning about relevance relations. Tech. Rep. R-53-L, UCLA Computer Science Department (1987)

Scalable Multi-Agent Reinforcement Learning with Cooperative Prioritized Sweeping

Eugenio Bargiacchi[1], Timothy Verstraeten[1], and Diederik M. Roijers[1,2]

[1] Vrije Universiteit Brussel, Brussels, Belgium
{eugenio.bargiacchi,timothy.verstraeten,diederik.roijers}@vub.be
[2] HU University of Applied Sciences, Utrecht, The Netherlands
diederik.yamamoto-roijers@hu.nl

Coordinating between a great number of agents in large-scale environments can be highly challenging due to the impact of the curse of dimensionality; learning policies and value functions in these settings is non-trivial due to exponentially sized joint state and joint action spaces. The task additionally requires tremendous amounts of data, which are not always available, and significantly increase the amounts of computational resources required.

We introduce *Cooperative Prioritized Sweeping* (CPS), a model-based algorithm with excellent scaling properties, which is able to coordinate between agents in a sample-efficient way and with a low computational cost.

CPS relies on the intuition that many large real-world cooperation problems benefit from locality, i.e. agents interact directly only with close-by neighbors and state features. CPS leverages this domain-knowledge information in the form of a *coordination graph* (CG) [Guestrin et al., 2002, Verstraeten et al., 2020], which specifies the way state features and agents are dependent on each other. CPS uses the CG to: efficiently learn a model of the environment, learn an approximate factorized value function and sample additional simulated environment interactions from the model to accelerate learning.

The main contribution of CPS lies in the way it uses the learned model to sample additional experience. Naively one could use the model to randomly generate new experience (the approach taken by Dyna-Q [Sutton, 1990]). However, in large environments, the curse of dimensionality quickly makes this approach ineffective, as most updates get applied to parts of the value function that do not need them. Instead, CPS's key idea is that we can improve sample-efficiency by detecting where updating the value function will be most effective. Recall the Bellman equation for the optimal value function:

$$V^\star(s) = R(s, a^\star) + \gamma \sum_{s'} T(s' |\ s, a^\star) V^\star(s') \tag{1}$$

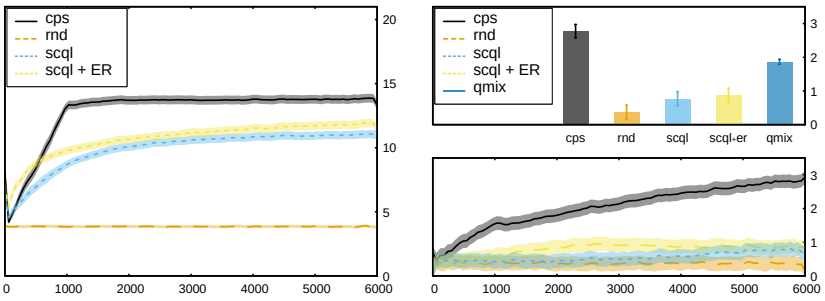
During learning, after each interaction with the environment we obtain new experience that can be used to update the value function. Equation 1 suggests that after each update for a particular state s', it is likely that the values for all its predecessor states s should be changed as well. The magnitude of the change for s will be proportional to (i) the temporal difference error of the initial update for s', and (ii) the probability of the transition $T(s' |\ s, a)$. Similar to the single-agent prioritized sweeping algorithm [Moore and Atkeson, 1993, Andre

et al., 1998] (PS), CPS summarizes this information through a *priority*, which is computed after each update of CPS's factored value function. In contrast to PS however, these priorities are also factorized, such that interesting, i.e., high-priority, joint states and actions can be efficiently identified. CPS's model of the environment is then sampled on these state-action pairs to generate synthetic experience and update the value function more quickly.

We perform several experiments to evaluate the empirical performance of CPS. We compare against against 4 benchmarks: a random policy as a naive approach, the factored LP planning algorithm on the ground truth MMDP model as the upper bound [Guestrin et al., 2002], Sparse Cooperative Q-learning (SCQL) with and without randomized experience replay [Kok and Vlassis, 2004], and QMIX [Rashid et al., 2018] as competing algorithms. The algorithms were implemented using the AI-Toolbox [Bargiacchi et al., 2020] and PYMARL [Samvelyan et al., 2019] frameworks. Figure 1 shows two typical results. Note that we do not plot the training for QMIX as its training took much longer than the other benchmarks. Figure 1(a) shows results in the SysAdmin [Guestrin et al., 2002] problem, using a torus topology with 10x10 agents. Figure 1(b) shows results in a randomly generated multi-agent setting with 15 agents.

We show that CPS is consistently faster learning and converges to better policies in the test benchmarks. Additionally, CPS is able to scale much better to large environments, both in time and compute resources. These properties make CPS a practical tool in tackling large-scale cooperation tasks.

Acknowledgements The authors would like to acknowledge FWO (Fonds Wetenschappelijk Onderzoek) for their support through the SB grant of Eugenio Bargiacchi (#1SA2820N), as well as the Flemish Government through the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.



(a) Torus setting with 100 agents. (b) Random setting with 15 agents.

Fig. 1: Histogram shows average per-timestep reward over 1000 timesteps for all policies, after training. Line plots show the mean and standard error of per-timestep reward of CPS and SCQL during training, compared against a random policy and the LP planning upper bound. All data is averaged over 100 runs.

References

[Andre et al., 1998] Andre, D., Friedman, N., and Parr, R. (1998). Generalized prioritized sweeping. In *Advances in Neural Information Processing Systems*, pages 1001–1007.

[Bargiacchi et al., 2020] Bargiacchi, E., Roijers, D. M., and Nowé, A. (2020). Aitoolbox: A c++ library for reinforcement learning and planning (with python bindings). *Journal of Machine Learning Research*, 21(102):1–12.

[Guestrin et al., 2002] Guestrin, C. E., Koller, D., and Parr, R. (2002). Multiagent planning with factored MDPs. In *NIPS 2002: Advances in Neural Information Processing Systems 15*, pages 1523–1530.

[Kok and Vlassis, 2004] Kok, J. R. and Vlassis, N. (2004). Sparse cooperative Q-learning. In *ICML 2004: Proceedings of the twenty-first international conference on Machine learning*, pages 61–68.

[Moore and Atkeson, 1993] Moore, A. W. and Atkeson, C. G. (1993). Prioritized sweeping: Reinforcement learning with less data and less time. *Machine learning*, 13(1):103–130.

[Rashid et al., 2018] Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. (2018). QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4292–4301.

[Samvelyan et al., 2019] Samvelyan, M., Rashid, T., De Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C.-M., Torr, P. H. S., Foerster, J., and Whiteson, S. (2019). The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043.

[Sutton, 1990] Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings 1990*, pages 216–224. Elsevier.

[Verstraeten et al., 2020] Verstraeten, T., Bargiacchi, E., Libin, P. J. K., Helsen, J., Roijers, D. M., and Nowé, A. (2020). Multi-agent Thompson sampling for bandit applications with sparse neighbourhood structures. *Nature Scientific Reports*, 10(1):6728.

Efficient Training of Robust Decision Trees Against Adversarial Examples

Daniël Vos and Sicco Verwer

Delft University of Technology, Delft, The Netherlands
{d.a.vos,s.e.verwer}@tudelft.nl

1 Introduction

Recently it has been shown that many machine learning models are vulnerable to adversarial examples: perturbed samples that trick the model into misclassifying them. Neural networks have received much attention but decision trees and their ensembles achieve state-of-the-art results on tabular data, motivating research on their robustness. Recently the first methods have been proposed to train decision trees and their ensembles robustly [4, 3, 2, 1] but the state-of-the-art methods are expensive to run.

We propose GROOT, an efficient algorithm for training robust decision trees. Like Chen et al. [3], we closely mimic the greedy recursive splitting strategy that traditional decision trees use and we score splits with the adversarial Gini impurity. We prove that the adversarial Gini impurity is concave with respect to the number of modifiable data points and use its analytical solution to compute the function in constant time. Our results show that GROOT trains trees 3 to 6 orders of magnitude faster than the state-of-the-art method TREANT [2] and trains random forests 100-1000 times faster than provably robust boosting [1].

2 GROOT: Growing Robust Trees

We introduce GROOT, an algorithm that trains decision trees that are robust against adversarial examples generated from a user-specified threat model. Like regular decision tree learning algorithms, GROOT runs in $\mathcal{O}(n \log n)$ time in terms of n samples. Similar to these algorithms, GROOT greedily makes splits according to a heuristic and while such strategies perform well in practice, they have no provable bound [5]. Where regular tree learning algorithms use the Gini impurity to score splits, GROOT uses the adversarial Gini impurity. This function represents the worst-case impurity after adversarial attacks, see Fig. 1.

3 Results

We evaluated the robustness of the algorithms on 13 tabular datasets by attacking all samples within a radius of 10% of the feature range in Fig. 2. GROOT decision trees and random forests on average perform as well as the existing

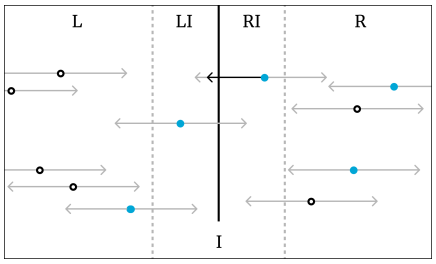


Fig. 1. We want to move samples from I over the threshold to maximize the weighted average of Gini impurities. Here we can move the single blue sample from RI into LI to maximize it.

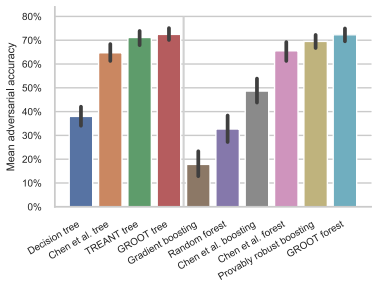


Fig. 2. Average adversarial accuracy over 13 structured datasets, GROOT trees and random forests achieve top results.

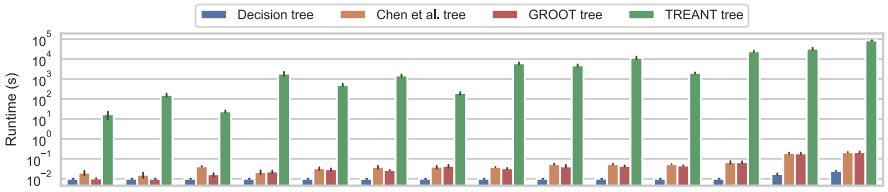


Fig. 3. Logarithmic training runtimes for single decision trees on different datasets. GROOT and Chen et al. run orders of magnitude faster than TREANT.

state-of-the-art works in trees and ensembles. To compare the efficiency of the algorithms, we plot the run times of each run in Figure 3 averaged over 5 data folds. All experiments ran on a single core of a Linux machine with 16 Intel Xeon CPU cores and 72GB of RAM total. Our results show that GROOT fits trees within seconds and scores as well as existing work against a box-shaped attack model. GROOT is available as a Scikit-learn compatible classifier[1].

References

1. Andriushchenko, M., Hein, M.: Provably robust boosted decision stumps and trees against adversarial attacks. arXiv preprint arXiv:1906.03526 (2019)
2. Calzavara, S., Lucchese, C., Tolomei, G., Abebe, S.A., Orlando, S.: Treant: Training evasion-aware decision trees. Data Mining and Knowledge Discovery pp. 1–31 (2020)
3. Chen, H., Zhang, H., Boning, D., Hsieh, C.J.: Robust decision trees against adversarial examples. In: ICML. pp. 1122–1131 (2019)
4. Kantchelian, A., Tygar, J.D., Joseph, A.: Evasion and hardening of tree ensemble classifiers. In: ICML. pp. 2387–2396 (2016)
5. Kearns, M.: Boosting theory towards practice: Recent developments in decision tree induction and the weak learning framework. In: Proceedings of the National Conference on Artificial Intelligence. pp. 1337–1339 (1996)

[1] https://github.com/tudelft-cda-lab/GROOT

Quick and Robust Feature Selection: the Strength of Energy-efficient Sparse Training for Autoencoders (Extended Abstract)* **

Zahra Atashgahi[1], Ghada Sokar[2], Tim van der Lee[3], Elena Mocanu[1], Decebal Constantin Mocanu[1,2], Raymond Veldhuis[1], and Mykola Pechenizkiy[2,4]

[1] EEMCS Faculty, University of Twente, the Netherland
[2] M&CS Faculty, Eindhoven University of Technology, the Netherlands
[3] EE Faculty, Eindhoven University of Technology, the Netherlands
[4] Faculty of Information Technology, University of Jyväskylä, Finland

1 Introduction

Feature selection, which identifies the most relevant and informative attributes of a dataset, has been introduced to address the challenges raised by the emerge of high-dimensional data [3]. Most existing feature selection methods are computationally inefficient; inefficient algorithms lead to high energy consumption, which is not desirable for devices with limited computational and energy resources. In [1], a novel and flexible method for unsupervised feature selection is proposed. This method, named "QuickSelection"[5], introduces the strength of the neuron in sparse neural networks as a criterion to measure the feature importance. When tested on several benchmark datasets, the proposed method is able to achieve the best trade-off of classification and clustering accuracy, running time, and memory usage, among widely used approaches for feature selection.

2 Proposed Method

QuickSelection is capable of selecting the most informative attributes of the data efficiently. The overview of the method is presented in Figure 1. This algorithm consists of two main phases: **1. Training Sparse DAE.** We use the ability of Denoising autoencoders (DAEs) to learn a robust representation of the data and select the most important features. We introduce for the first time sparse training in the world of denoising autoencoders, and we name the newly introduced model sparse denoising autoencoder (sparse DAE). We train the sparse DAE with the Sparse Evolutionary Training (SET) [4] algorithm to keep the number of parameters low during the training. **2. Feature Selection.** In the second phase, we use the trained network to derive the hierarchical importance of the

* This research has been partly funded by the NWO EDIC project.
** The full paper corresponding to this abstract has been accepted for publication in the Machine Learning Journal (ECML-PKDD 2022 Journal Track)
[5] The code is available at: `https://github.com/zahraatashgahi/QuickSelection`

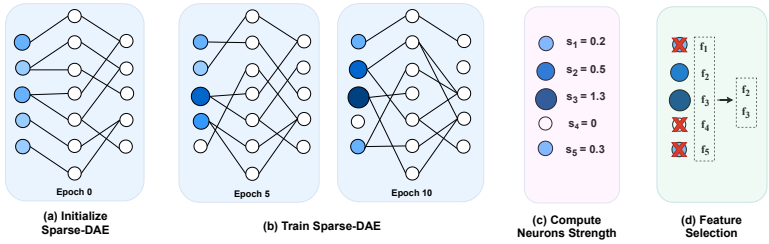


Fig. 1. A high-level overview of the proposed method, "QuickSelection".

features. We select the most important features of the data based on the weights of their corresponding input neurons of the trained sparse DAE. Inspired by node strength in graph theory [2], we determine the importance of each neuron based on its *strength*. We estimate the strength of each neuron by the summation of absolute weights of its outgoing connections. We select the features corresponding to the neurons with K largest strength values as the K important features.

Results. In order to verify the validity of our proposed method, we carry out several experiments to measure its performance in terms of the running time, memory requirement, clustering accuracy, and classification accuracy. To analyze the trade-off of the methods between accuracy and efficiency, we compute a ranking-based score (Figure 2): on several datasets and for several values of K, we rank the methods based on the aforementioned metrics. Then, we give a score of 1 to the best and second-best performers. As can be seen in Figure 2, our proposed method can achieve the best trade-off between accuracy, running time, and memory usage, among all the considered methods.

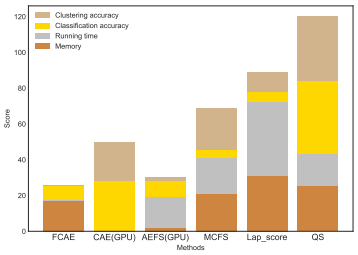


Fig. 2. Feature selection comparison using a ranking-based score.

3 Concluding Remarks

In this paper [1], a novel method (QuickSelection) for energy-efficient unsupervised feature selection has been proposed. We introduced *neuron strength* as a metric to measure the importance of the input neurons in a sparse neural network. By adopting this metric in a sparsely connected denoising autoencoder, we are able to derive the importance of all input features simultaneously. By using sparse layers instead of dense ones from the beginning, the number of parameters drops significantly. As a result, QuickSelection requires much less memory, computational resources, and energy consumption than its competitors. This will not only save the energy costs of processing high-dimensional data but also will ease the challenges of high energy consumption imposed on the environment.

References

1. Atashgahi, Z., Sokar, G., van der Lee, T., Mocanu, E., Mocanu, D.C., Veldhuis, R., Pechenizkiy, M.: Quick and robust feature selection: the strength of energy-efficient sparse training for autoencoders. Accepted at Machine Learning Journal (ECML-PKDD 2022 Journal Track) preprint arXiv:2012.00560 (2020)
2. Barrat, A., Barthelemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. Proceedings of the National Academy of Sciences **101**(11), 3747–3752 (2004)
3. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Computers & Electrical Engineering **40**(1), 16–28 (2014)
4. Mocanu, D.C., Mocanu, E., Stone, P., Nguyen, P.H., Gibescu, M., Liotta, A.: Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. Nature Communications **9**(1), 2383 (2018)

Assessing the Quality of Online Reviews Using Formal Argumentation Theory

Davide Ceolin, Giuseppe Primiero, Jan Wielemaker, Michael Soprano

[1] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
{davide.ceolin, j.wielemaker}@cwi.nl
[2] University of Milan, Milan, Italy giuseppe.primiero@unimi.it
[3] University of Udine, Udine, Italy michael.soprano@uniud.it

Review scores collect users' opinions in a simple and intuitive manner. However, review scores are also easily manipulable, hence they are often accompanied by explanations. A substantial amount of research has been devoted to ascertaining the quality of reviews, to identify the most useful and authentic scores through explanation analysis. In this paper, we advance the state of the art in review quality analysis. We introduce a rating system to identify review arguments and to define an appropriate weighted semantics through formal argumentation theory. We introduce an algorithm to construct a corresponding graph, based on a selection of weighted arguments, their semantic similarity, and the supported ratings. Such an algorithm identifies tokens in corpora of reviews, and then clusters them according to their similarity. Token similarity is measured using the Word Mover distance [4], since it allows measuring semantic similarity of short items of text. Attacks are defined between tokens when they belong to conflicting reviews (i.e., to reviews which scores are different). Such attacks are weighted on the readability level of the reviews and on the importance of the token in the review. Potential arguments are considered as stronger when they belong the most readable reviews, and when their importance in the review is high. As a readability measure, we use the Flesch Kincaid Reading Ease measure [3]. This formula provides reliable scores between 100 (text understandable by 5th graders) and 0 (texts understandable by professionals). Other readability measures will be tested in the future. As a measure of the importance of the possible arguments, we employ the textRank algorithm [6].

We provide an algorithm to identify the model of such an argumentation graph, maximizing the overall weight of the admitted nodes and edges. We evaluate these contributions on the Amazon review dataset by McAuley et al. [5], by comparing the results of our argumentation assessment with the upvotes received by the reviews. Also, we deepen the evaluation by crowdsourcing a multidimensional assessment of reviews and comparing it to the argumentation assessment. We use a dedicated crowdsourcing platform where we ask crowd contributors to assess the quality of reviews according to seven dimensions of quality (truthfulness, reliability, neutrality, comprehensibility, precision, completeness, informativeness). These dimensions are based on literature [1] and allow evaluating the quality of dimensions according to different and possibly independent aspects. Lastly, we perform a user study to evaluate the explainability of our method. Our method achieves three goals: (1) it identifies reviews that are considered useful when looking at their number of upvotes; (2) when deepening the analysis on the quality of the reviews that are accepted on the basis of argumentation reasoning, we can observe that, in particular, they are considered as comprehensible and truthful; and

(3) our user study shows that our approach provides a comprehensible explanation of review quality assessments.

This extended abstract is based on a paper published at ICWE [2].

References

1. Ceolin, D., Noordegraaf, J., Aroyo, L.: Capturing the ineffable: Collecting, analysing, and automating web document quality assessments. In: Proceedings of EKAW. p. 83–97. Springer (2016)
2. Ceolin, D., Primiero, G., Wielemaker, J., Soprano, M.: Assessing the quality of online reviews using formal argumentation theory. In: Brambilla, M., Chbeir, R., Frasincar, F., Manolescu, I. (eds.) Web Engineering. pp. 71–87. Springer International Publishing, Cham (2021)
3. Kincaid, J., Fishburne, R., Rogers, R., Chissom, B.: Derivation of new readability formulas for navy enlisted personnel. research branch report 8–75. Tech. rep., Chief of Naval Technical Training: Naval Air Station Memphis (1975)
4. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: Proceedings of ICML. p. 957–966. JMLR.org (2015)
5. McAuley, J.J., Targett, C., Shi, Q., van den Hengel, A.: Image-based recommendations on styles and substitutes. In: Proceedings of SIGIR. pp. 43–52. ACM (2015)
6. Mihalcea, R., Tarau, P.: TextRank: Bringing order into text. In: Proceedings of EMNLP. pp. 404–411. ACL (2004)

Agent-Based Simulation of Short-Term Peer-to-Peer Rentals: Evidence from the Amsterdam Housing Market (Article Abstract)*

Neil Yorke-Smith[0000−0002−1814−3515]
n.yorke-smith@tudelft.nl

Delft University of Technology, The Netherlands

Abstract. The full article published in Environment and Planning B studies the effect of a range of possible municipal policy measures on the peer-to-peer short-term rental market. The case study is the city of Amsterdam. A spatial agent-based simulation indicates that more lower income citizens remain in the city centre when regulation of the market is stronger, and that banning the touristic market restrains the overall increase in house prices, compared to the business-as-usual scenario. However, the feasibility of enforcement of regulation, and its libertarian consequences, must be considered.

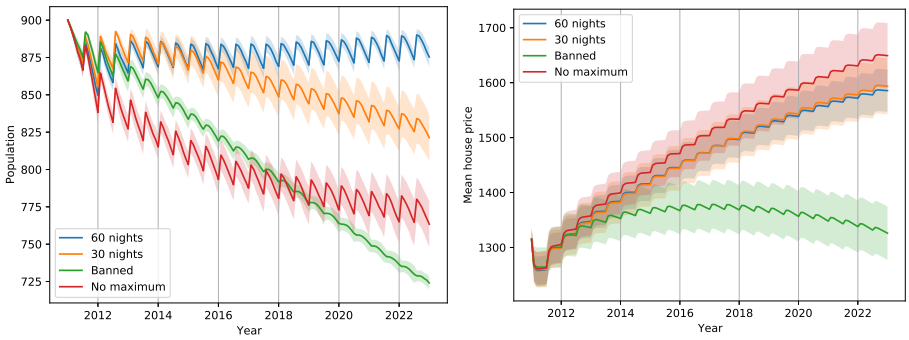
1 Motivation and Approach

The full article by Overwater and Yorke-Smith [6] recognises that gentrification, displacement and social exclusion are hot topics of debate in the city of Amsterdam, the Netherlands. A current phenomena is short-term rentals of private homes. In its peer-to-peer form, this phenomena has grown sharply, facilitated by services such as Airbnb. Its growth has caused controversies among communities in touristic areas of Amsterdam, since it contributes to a changed social fabric, increased housing prices and overall gentrification [3, 11]. In the Netherlands and elsewhere, municipal and national policy makers are interested to regulate short-term rentals [2].

The article's methodological lens to study the 'Airbnb effect' on Amsterdam – and to provide insights into qualitative policy effects on the regulation of short-term rentals – is a micro-level agent-based simulation. The agent-based model (ABM) developed is grounded in data. In contrast to Vinogradov et al. [10] the model is geographically accurate, and is based on Smith's rent-gap hypothesis rather than a real estate market model.

The spatial agent-based model captures two types of agents: city residents and visiting tourists. The article builds upon an extant ABM of urban residential dynamics [7, 9] that combines Smith's rent-gap theory [8] and Axelrod's cultural

* This is an extended abstract of [6].



(a) Resident population (b) House prices

Fig. 1: Comparison of four regulation scenarios

exchange theory [1], adapting it significantly to model the touristic rental market. It captures private and social housing, and residents' propensity to rent their dwelling, under the cases where it is and is not permitted by contract. While calibrated on data from Amsterdam and the Dutch legal setting, the modelling approach presented in the article is generic.

2 Results and Discussion

Simulation analysis shows that the tourism market has caused considerable changes in housing prices and population development. As an example of the article's results, Figure 1 shows population and house prices under four regulation scenarios: no regulation, 60 nights rental per property per year, 30 nights, and a complete ban. The simulation proceeds from a start year of 2011 for ten years.

The article finds that more lower income citizens will live in the city when regulation of the market is stronger. Banning the touristic market restrains the overall increase in house prices, compared to the business-as-usual scenario. However, the feasibility of enforcement of regulation [5], and its libertarian consequences [4], must be considered. Indeed, a complete ban would not align with the 'tolerant' Dutch culture. Thus the main conclusion for the case of Amsterdam is that tighter limits, on the amount of nights a property can be listed on Airbnb, is preferable to an outright ban.

A future enrichment will be to survey residents in order to better characterise their attitude to restrictions and touristic rentals, and their propensity to ignore rules and contracts [5]. The interplay of the short-term and long-term rental markets deserves further study using agent-based simulation.

Acknowledgements Thanks to Jan Kwakkel and Stephen van der Laan, and the E&PB reviewers. This research was partially supported by TAILOR, a project funded by EU Horizon 2020 programme grant number 952215.

References

[1] Axelrod, R.: The Complexity of Cooperation. Princeton University Press (1997)

[2] Furukawa, N., Onuki, M.: The design and effects of short-term rental regulation. Current Issues in Tourism (2019), e-print

[3] Henley, J.: Overtourism in Europe's historic cities sparks backlash. The Guardian (25 jan 2020)

[4] Kadi, J., Musterd, S.: Housing for the poor in a neo-liberalising just city: Still affordable, but increasingly inaccessible. Tijdschrift voor Economische en Sociale Geografie **106**(3), 246–262 (2015)

[5] Leshinsky, R., Schatz, L.: "I don't think my landlord will find out": Airbnb and the challenges of enforcement. Urban Policy and Research **1146**, 1–12 (2018), ISSN 0811-1146

[6] Overwater, A., Yorke-Smith, N.: Agent-based simulation of short-term peer-to-peer rentals: Evidence from the amsterdam housing market. Environment and Planning B: Urban Analytics and City Science p. 23998083211000747 (2021), https://doi.org/10.1177/23998083211000747

[7] Picascia, S., Yorke-Smith, N.: Towards an agent-based simulation of housing in urban Beirut. In: Post-proceedings of AAMAS'16 Workshop on Agent Based Modelling of Urban Systems (ABMUS'16), LNCS, vol. 10051, pp. 3–20, Springer (2017)

[8] Smith, N.: Gentrification and capital: theory, practice and ideology in society hill. Antipode **11**(3), 24–35 (1979)

[9] Termos, A., Picascia, S., Yorke-Smith, N.: Agent-based simulation of west asian urban dynamics: Impact of refugees. Journal of Artificial Societies and Social Simulation **24**(1), 2 (2021), https://doi.org/10.18564/jasss.4472

[10] Vinogradov, E., Leick, B., Kivedal, B.K.: An agent-based modelling approach to housing market regulations and Airbnb-induced tourism. Tourism Management **77**, 104004 (2020)

[11] Wachsmuth, D., Weisler, A.: Airbnb and the rent gap: Gentrification through the sharing economy. Environment and Planning A: Economy and Space **50**, 1147–1170 (2018)

Learning 2-opt Local Search from Demonstrations

Paulo da Costa, Yingqian Zhang, Alp Akcay, and Uzay Kaymak

Eindhoven University of Technology, 5612 AZ Eindhoven, Netherlands
{p.r.d.oliveira.da.costa, yqzhang, a.e.akcay, u.kaymak}@tue.nl

Abstract. Deep reinforcement learning (RL) has achieved high success in solving routing problems. However, state-of-the-art deep RL approaches require a considerable amount of data before they reach reasonable performance. This limits the applicability of these methods to many real-world instances. This work studies a setting where the agent can access data from suboptimal heuristics for the traveling salesman problem. The agent has access to demonstrations from 2-opt improvement policies and our goal is to learn policies that can surpass the quality of the demonstrations requiring fewer samples than pure RL. We propose to first learn policies via behavior cloning, leveraging a small set of demonstrations. Afterwards, we combine on policy and value approximation updates to improve performance. We show that our method learns good policies in a shorter time and using less data than classical policy gradient. Moreover, it performs similarly to other state-of-the-art deep RL approaches.

Keywords: Deep Reinforcement Learning · Combinatorial Optimization · Traveling Salesman Problem.
Acknowledgments: This research is funded by NWO Big data: Real Time ICT for Logistics, project number 628.009.012
Publication: The full paper of this abstract has been accepted at the 2021 International Joint Conference in Neural Networks.

1 Introduction

The traveling salesman problem (TSP) is a well-known combinatorial optimization (CO) problem where the aim is to find an optimal tour that visits n locations once and returns to the origin. The TSP is NP-hard, [1] and solving large TSP instances optimally can be impractical due to high computational costs. For that reason, several (meta)heuristics have been proposed the problem. However, these rely on expert knowledge and may perform poorly if the regularity of problem instances are not considered during development.

2 Methods

Recent reinforcement learning (RL) methods aim at learning better policies for such problems, exploiting the regularity of problem instances. However, RL

methods require many samples and high computational time to learn policies that can compete with metaheuristics. Thus, this work proposes to use demonstration data from expert heuristics and learn via behavior cloning (BC). Because expert heuristics can be sub-optimal, our approach also considers a second learning stage in which RL is employed to surpass the quality of the heuristic. Our proposed method focuses on the classical improvement heuristic based on a 2-edge swap change (2-opt) to a TSP solution. We consider as expert demonstrations the greedy version of the heuristics that select the swap that leads to best improvement (BI) in cost and the one that chooses the first improvement (FI). Our approach can achieve similar results to previous RL methods after learning from demonstrations and a few interactions of RL training, requiring lower training times and sample complexity. In the experiments, we collect expert demonstrations from FI and BI heuristics, extract a policy from demonstrations and perform policy gradient [2] updates over online environment interactions to improve upon the expert policies.

3 Results

We learn policies for TSP instances with 20, 50 and 100 nodes, and depict the optimality gap for 10,000 test instances in Table 1. The results show that we can learn effective *early* policies that decrease the optimality gap over the training epochs and can approximate the performance of the previous methods for the same task learning solely via RL.

Table 1: S: Number of samples. *Type*: Solver, **SL**: Supervised Learning, **BC**: Behavior Cloning, **RL**: Reinforcement Learning. @{iterations of PG}

Method	Type	TSP20			TSP50			TSP100		
		Cost	Gap	S ($\times 10^8$)	Cost	Gap	S ($\times 10^8$)	Cost	Gap	S ($\times 10^8$)
PG@1	RL	7.62	98.72%	0.01	14.08	147.30%	0.01	33.66	333.50%	0.01
PG@5	RL	4.04	5.26%	0.05	7.59	33.30%	0.05	11.06	42.50%	0.05
PG@20	RL	3.85	0.21%	0.21	5.94	4.28%	0.21	8.56	10.29%	0.21
PG@200	RL	3.84	0.01%	2.05	5.71	0.30%	2.05	7.89	1.61%	2.05
BC+PG@1	BC, RL	3.88	1.17%	0.01	6.28	10.29%	0.01	8.86	14.15%	0.01
BC+PG@5	BC, RL	3.84	0.07%	0.05	5.84	2.61%	0.05	8.47	9.02%	0.05
BC+PG@20	BC, RL	3.84	0.02%	0.21	5.74	0.86%	0.21	8.07	3.98%	0.21
BC+PG@200	BC, RL	**3.84**	**0.00%**	2.05	5.71	0.21%	2.05	7.87	1.41%	2.05

References

1. Papadimitriou, C.H.: The euclidean travelling salesman problem is np-complete. Theoretical Computer Science **4**(3), 237–244 (1977)
2. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3-4), 229–256 (1992)

SpaceNet: Make Free Space For Continual Learning (Extended Abstract) ⋆

Ghada Sokar[1], Decebal Constantin Mocanu[1,2], and Mykola Pechenizkiy[1]

[1] Eindhoven University of Technology, The Netherlands
[2] University of Twente, The Netherlands

1 Introduction

Continual learning aims to build intelligent agents that can continuously learn new tasks over time while preserving the old learned knowledge. Ideally, the agent should continually learn without adding a huge computational and memory overhead to learn a new task or remember the old ones. The main challenge in this paradigm is catastrophically forgetting previous tasks when the model is optimized for a new one [6]. Existing methods mitigate this problem at the expense of increasing the model capacity [12,9] or replaying the old samples [8]. This hinders its applicability to real-world applications where the old data might be inaccessible and computation and memory efficiency are required. To address these limitations, we proposed SpaceNet [11] [1] a new architectural-based strategy that utilizes the available *fixed-capacity* of the model efficiently. We harness the significant redundancy of deep neural networks [2] and learn each task in a compact space using dynamic sparse training. SpaceNet learns *semi-distributed sparse* representation for each task. This representation has two key advantages: (1) it reduces the interference between tasks and (2) it leaves free neurons for future tasks without adding extra computation and memory overhead.

2 Proposed Method

SpaceNet is a brain-inspired method that mimics the high sparse activity in the brain [1,4]. It is motivated by the recent success of dynamic sparse training methods in achieving a similar performance of dense neural networks using high sparse networks [7,5]. An overview of SpaceNet is illustrated in Figure 1. We dynamically train each task from scratch using a *sparse sub-network*. SpaceNet consists of three main phases. (1) Sparse connections allocation between the *free* neurons in each layer. (2) Dynamic sparse training. During learning each task, the sparse topology is optimized for paying more attention to the useful neurons for the current task. In particular, the distribution of the sparse connections is adaptively changed and compacted in the most important neurons for the current task through drop-and-grow cycles (Figure 2).

⋆ The full paper has been published in Elsevier Neurocomputing, Volume 439, 2021, Pages 1-11. https://doi.org/10.1016/j.neucom.2021.01.078.

[1] The code is available at: https://github.com/GhadaSokar/SpaceNet

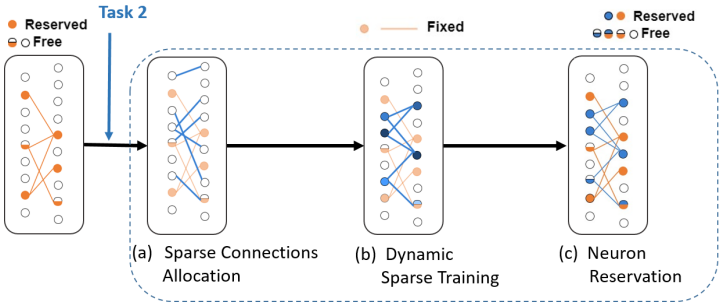


Fig. 1: An overview of our proposed method SpaceNet. The figure shows the learning process of Task 2 in the sequence. SpaceNet consists of three main steps. (a) **Sparse connections allocation** for the current task between the *free* neurons that are not reserved by previously learned tasks. (b) **Dynamic sparse training** in which the weights and the sparse topology are optimized simultaneously for the current task. (c) **Neuron reservation** of the important neurons for the current task and removing them from the *free* list of neurons.

(3) Neuron reservation. After training, a fraction of the important neurons for the current task is reserved and can not be used by other tasks. This results in *sparse representations* for each task which reduces the interference between the tasks, hence forgetting. Table 1 shows the performance of SpaceNet compared to other strategies. As shown in the table, SpaceNet outperforms the regularization and architectural methods by a big margin. It also achieves promising results compared to the rehearsal strategy given that SpaceNet does not use previous data.

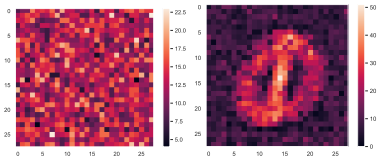


Fig. 2: Visualization of the number of weights connected to each input neurons for Task 1, digits (0,1), in the Split MNIST benchmark [13] before (left) and after (right) training.

Table 1: Accuracy (ACC) on split MNIST [13] using different approaches.

Strategy	Method	ACC (%)
Regularization	EWC [3]	20.01 ± 0.06
Rehearsal	DGR [10]	90.79 ± 0.41
	SpaceNet-Rehearsal	**95.08** ± 0.15
Architectural	DEN [12]	56.95 ± 0.02
	SpaceNet	**75.53** ± 1.82

3 Conclusion

In this paper [11], we proposed an architectural-based strategy to continually learn a set of tasks sequentially without forgetting. We introduced a dynamic sparse training algorithm to train each task to produce sparse representation. By learning these sparse representations, we managed to reduce the forgetting in previous tasks without replaying previous data. We showed that we can accumulate knowledge over time while preserving the old one with a negligible memory and computation overhead.

References

1. Attwell, D., Laughlin, S.B.: An energy budget for signaling in the grey matter of the brain. Journal of Cerebral Blood Flow & Metabolism **21**(10), 1133–1145 (2001)
2. Denil, M., Shakibi, B., Dinh, L., Ranzato, M., de Freitas, N.: Predicting parameters in deep learning. In: Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2. pp. 2148–2156 (2013)
3. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
4. Lennie, P.: The cost of cortical computation. Current biology **13**(6), 493–497 (2003)
5. Liu, S., Van der Lee, T., Yaman, A., Atashgahi, Z., Ferraro, D., Sokar, G., Pechenizkiy, M., Mocanu, D.C.: Topological insights in sparse neural networks. arXiv preprint arXiv:2006.14085 (2020)
6. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)
7. Mocanu, D.C., Mocanu, E., Stone, P., Nguyen, P.H., Gibescu, M., Liotta, A.: Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. Nature communications **9**(1), 2383 (2018)
8. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
9. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
10. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: Advances in Neural Information Processing Systems. pp. 2990–2999 (2017)
11. Sokar, G., Mocanu, D.C., Pechenizkiy, M.: Spacenet: Make free space for continual learning. Neurocomputing **439**, 1–11 (2021)
12. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. In: International Conference on Learning Representations (2018)
13. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International Conference on Machine Learning. pp. 3987–3995. PMLR (2017)

Unsupervised Online Grounding for Social Robots (Extended Abstract)*

Oliver Roesler[1] and Elahe Bagheri[2]
[1] Artificial Intelligence Lab, Vrije Universiteit Brussel, Brussels, Belgium
oliver@roesler.co.uk
[2] Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel and Flanders Make, Brussels, Belgium elahe.bagheri@vub.be

1 Introduction

Robots that incorporate social norms in their behaviors are seen as more supportive and friendly. Since it is impossible to manually specify the most appropriate behavior for all possible situations, robots need to be able to learn it through trial and error, by observing interactions between humans, or by utilizing theoretical knowledge available in natural language. In contrast to the former two approaches, the latter has not received much attention because understanding natural language is non-trivial and requires proper grounding mechanisms to link words to corresponding perceptual information. Previous grounding studies have mostly focused on grounding of concepts relevant to object manipulation [1,4], while grounding of more abstract concepts relevant to the learning of social norms has so far not been investigated.
In this paper, we present an unsupervised cross-situational learning based online grounding framework to ground emotion types, emotion intensities and genders through their corresponding concrete representations, which represent sets of invariant perceptual features obtained through an agent's sensors that are sufficient to distinguish percepts belonging to different concepts [3], extracted from audio with the help of deep learning. The proposed framework is evaluated through a simulated human-agent interaction experiment in which the agent listens to the speech of different people and receives at the same time a natural language description, describing the gender of the observed person as well as the experienced emotion. Furthermore, the proposed framework is compared to a Bayesian grounding framework that has been employed in several previous studies to ground words through a variety of different percepts [1,4].

2 System Overview

The employed grounding framework consists of three parts: (1) Perceptual feature extraction component, which extracts audio features from video using openEAR [2], (2) Perceptual feature classification component, which uses deep neural networks to obtain concrete representations of perceptual features, (3) Language grounding component, which identifies auxiliary words, i.e. words that have no corresponding concrete representations, and creates mappings from non-auxiliary words to corresponding concrete representations using cross-situational learning.

* This is an extended abstract of Roesler and Bagheri [5].

3 Results

The obtained results show that the framework is able to identify auxiliary words and ground non-auxiliary words, including synonyms, referring to abstract concepts through their corresponding emotion types, emotion intensities and genders. Furthermore, they illustrate that the grounding algorithm employed by the proposed framework depends on the accuracy of the used concrete representations, which are in this study obtained through deep learning, but does not require perfectly accurate representations because the framework is already able to obtain all correct mappings, if the accuracy of the concrete representations is on average only around 85% for all considered modalities. Additionally, the proposed framework outperformed the baseline framework in terms of the accuracy of the obtained groundings as well as its ability to learn new groundings and continuously update existing groundings during interactions with other agents and the environment, which is essential when considering real-world deployment. Finally, the framework is also more transparent, due to the creation of explicit mappings from words to concrete representations.

4 Conclusion

The proposed framework allowed identification of auxiliary words and grounding of abstract concepts, like emotion types, emotion intensities and genders, through their corresponding concrete representations in an online manner using cross-situational learning. In future work, we will integrate the framework with a knowledge representation to explore the utilization of abstract knowledge to increase the sample-efficiency of the grounding mechanism as well as the accuracy of the obtained groundings, and enable agents to reason about the world with the help of an abstract but grounded world model.

References

1. Aly, A., Taniguchi, T.: Towards Understanding Object-Directed Actions: A Generative Model for Grounding Syntactic Categories of Speech through Visual Perception. In: IEEE International Conference on Robotics and Automation (ICRA). Brisbane, Australia (May 2018)
2. Eyben, F., Wöllmer, M., Schuller, B.: OpenEAR - Introducing the Munich Open-Source Emotion and Affect Recognition Toolkit. In: Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops. Amsterdam, Netherlands (September 2009)
3. Harnad, S.: The Symbol Grounding Problem. Physica D **42**, 335–346 (1990)
4. Roesler, O.: Unsupervised Online Grounding of Natural Language during Human-Robot Interaction. In: Second Grand Challenge and Workshop on Multimodal Language at ACL 2020. Seattle, USA (July 2020)
5. Roesler, O., Bagheri, E.: Unsupervised Online Grounding for Social Robots. Robotics **10**(2) (April 2021). https://doi.org/https://doi.org/10.3390/robotics10020066

Posters and demonstrations

BNAIC/BeneLearn proceedings
November 10–12, 2021
Belval, Esch-sur-Alzette (Luxembourg)

Play the Reinforcement Learning Agent

Hélène Plisnier, Alessandro Fasano, and Ann Nowé

Vrije Universiteit Brussels
helene.plisnier@vub.be
https://youtu.be/CP4HPBzsmtU

Abstract. In the past few decades, artificial intelligence has gained an increasing amount of interest from the general public. Accompanying this interest, comes expectations of how sophisticated AI methods and their abilities are, often without a proper understanding of how they actually work. This demonstration is meant to give non-expert participants an idea of the view an RL agent has of its environment. We invite a volunteer to take the place of a standard RL agent and try learning the task solely based on information that would be available in a typical RL setting. The purpose of this demonstration is to illustrate how unintuitive an RL agent's perspective of its environments is from a human point of view, and hence how limited its understanding of the task it is learning is. By establishing this idea in non-experts minds, we hope to debunk certain inaccurate assumptions people may have about AI technologies, specifically RL in this case.

Keywords: Reinforcement Learning · Transparency · Volunteer-Driven Demonstration

1 Introduction

Reinforcement Learning [3] (RL) is an Artificial Intelligence method, in which an agent learns to perform a task from scratch by repetitively interacting with its environment. At each timestep, the agent chooses and executes one of the actions at its disposal based on some state information, then receives a reward or punishment and goes on to the next state. RL methods consist in a promising approach to make robots easier to deploy in human-populated spaces, such as industries, offices and homes.

However, for robots executing RL algorithms to be accepted and allowed in human spaces, and to comply to (current and future) AI Transparency [1] and Explanability regulations [1] their decision-making processes must be made clear for human users [6, 4, 2, 5]. An important component of the RL process is the way RL agents "perceive" and "understand" their environment. That kind of information is often restricted to experts in RL, who are used to design RL algorithms and test them on different environments. Our demonstration has been

[1] Such as the General Data Protection Regulation: https://eur-lex.europa.eu/eli/reg/2016/679/oj

developed with the intent to give non-expert participants a *feel* of the reasoning used by an RL agent. Such demonstration can help users level their expectations of the capabilities of AI technologies, and help them shape an informed mental representation of these technologies. In this paper, we present a demonstration involving a volunteer, whom is put in the position of a reinforcement learner, with access to a number of actions, some state information, and reward signals in the form of colored lights (see Figure 1). Without knowing it, the volunteer is controlling a simulated hook used to move a container onto a ship; this graphical representation of the task is not shown to the volunteer during the demonstration, but is visible to accompanying people, allowing them to follow the volunteer's progress.

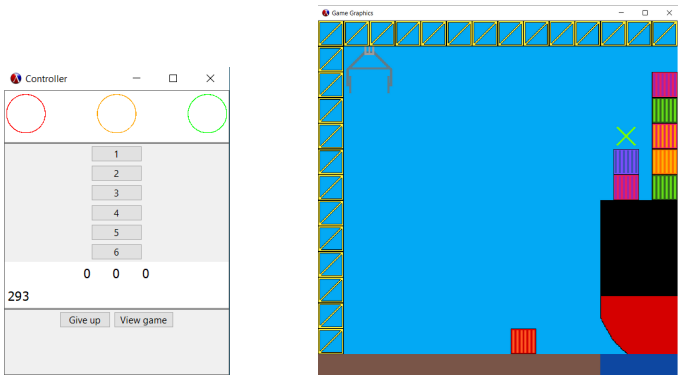


Fig. 1. *Left:* Controller window provided to the volunteer. *Right:* Graphical representation of the task. The volunteer is not shown this window during the demonstration.

Acknowledgement

The first author is funded by the Science Foundation of Flanders (FWO, Belgium) as 1SA6619N Applied Researcher.

References

1. Felzmann, H., Villaronga, E.F., Lutz, C., Tamò-Larrieux, A.: Transparency you can trust: Transparency requirements for artificial intelligence between legal norms and contextual concerns. Big Data & Society **6**(1), 2053951719860542 (2019)
2. Hagras, H.: Toward human-understandable, explainable ai. Computer **51**(9), 28–36 (2018)
3. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2018)

4. Thomaz, A.L., Breazeal, C.: Transparency and socially guided machine learning. In: 5th Intl. Conf. on Development and Learning (ICDL) (2006)
5. Vallor, S., Bekey, G.A.: Artificial intelligence and the ethics of self-learning robots (2017)
6. Weller, A.: Challenges for transparency (2017)

A Design Pattern Language for Hybrid Intelligent Teams

Mani Tajaddini[1][0000−0001−9038−3524], Willem-Paul Brinkman[1][0000−0001−8485−7092], Annette ten Teije[2][0000−0002−9771−8822], and Mark Neerincx[1,3][0000−0002−8161−5722]

[1] Interactive Intelligence Group, Delft University of Technology, The Netherlands
{w.p.brinkman,m.a.neerincx,m.tajaddini}@tudelft.nl
[2] Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands
annette.ten.teije@vu.nl
[3] TNO Human Factors, Soesterberg, The Netherlands
mark.neerincx@tno.nl

Abstract. The field of Hybrid Intelligence (HI) is like a vast land with many tribes that speak different languages. Our goal is to develop a *lingua franca* to unify the peoples of the HI land. We expect our language to facilitate documentation and communication of research results and thus collaboration among various HI fields by making use of design patterns describing human-AI interactions.

Keywords: Design pattern language · Hybrid intelligence · Human-AI interaction.

1 Introduction

The premise of this project is to create a language whose words are design patterns in Hybrid Intelligence (i.e., HI) design. These patterns describe a configuration of machine and human agents which are designed to carry out a particular task in a particular set of circumstances [1, 4]. The difficulty of such an endeavor lies in the fact that the HI field is like a vast land with many tribes (i.e., groups of scientists and engineers from many different disciplines) working in it. These tribes usually speak different languages and work in diverse contexts and have diverse backgrounds. This causes dispersion in communication between various tribes and through time. Difficulties arise in integrating one tribe's work in another tribe's project, in communicating one tribe's findings in a certain field to another field, in getting two or more tribes to collaborate on a project, and even to deliver one tribe's work through time so that future projects can benefit from it.

We would like to facilitate communication throughout the HI land by creating a *lingua franca* which all the various peoples of the HI land can speak. To do this, a strategy is needed to document the experiences of designers in various HI fields by extracting useful patterns from those experiences. In effect, future designers will not need to reinvent the wheel and will have a vocabulary and a framework that will guide them in their designing efforts.

2 Development and Evaluation

What we need in order to build a language is a vocabulary and a grammar (which is some kind of structure). The vocabulary is the set of design patterns and most design pattern languages stop there. They are just a catalogue or a dictionary of design patterns. However, having the same vocabulary does not necessarily mean mutual intelligibility between two languages. To have a universal language we also need a common grammar. The grammar is a formalism (e.g., context-free grammar, combinators, lambda calculus, category theory, etc.) that structures the vocabulary and dictates how to compose different design patterns within different levels and between different levels of abstraction.

As a starting point, we should identify design patterns through observing how people working in HI think and solve their problems. In representing design patterns we should focus on how these patterns compose so that we can superimpose a structure as the language's grammar. Our effort will be geared toward generating as many useful syntactically possible combinations of our atomic patterns as possible [2, 3]. We expect the pattern language to have both an easy-to-access graphical notation and a formal representation that can be manipulated by computer-tools (e.g., editor, validator, search tool, configurator, etc). One very idealistic end product to imagine is a Domain Specific Language accessible to both humans and machines which could be used to design an Integrated Development Environment.

In evaluating how successful the design pattern language is, we can take into account the following. Firstly, the employed design patterns should be valid, meaning they have to be instantiated in a concrete context to see how far the system behaves according to the pattern. To do this, in certain cases, tests involving simulations to address the size, diversity, and dynamics of the human and artificial cognitive processes can be carried out. Furthermore, the design pattern language should be as complete as possible or viable, meaning it must be able to describe as many as the HI application scenarios which are useful for the users. However, the most important questions to ask in evaluating the language should address researchers' and engineers' performance in reliably instantiating an HI design pattern into a solution for their situated problem. Questions like how easy it is to understand the idea expressed in a design pattern; how easy it is to find a design pattern expressing a certain idea; how well the language expresses the important properties of an idea, such as its scope or impact; and is it possible to compare design patterns or ideas therein using the pattern language. It is also very important that our language be dynamic, meaning designers must be able to add to it and change it as the research field develops.

3 Concluding Remarks

The above paragraphs sketch out a plan for constructing a design pattern language. However, to begin with, we must understand what a design pattern means in the field of HI; which design patterns are available in this field; and what efforts have been made in formalizing design pattern languages. Therefor, we have

set out to write two review papers on design patterns in HI and on methods of design pattern formalization. The future plan consists of incrementally discovering and formalizing extant design patterns in HI and identifying gaps therein.

References

1. Akata, Z., Balliet, D., De Rijke, M., Dignum, F., Dignum, V., Eiben, G., Fokkens, A., Grossi, D., Hindriks, K., Hoos, H., et al.: A research agenda for hybrid intelligence: augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. Computer **53**(8), 18–28 (2020)
2. van Bekkum, M., de Boer, M., van Harmelen, F., Meyer-Vitali, A., Teije, A.t.: Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases. arXiv preprint arXiv:2102.11965 (2021)
3. Van Harmelen, F., Teije, A.t.: A boxology of design patterns for hybrid learning and reasoning systems. arXiv preprint arXiv:1905.12389 (2019)
4. van der Waa, J., van Diggelen, J., Siebert, L.C., Neerincx, M., Jonker, C.: Allocation of moral decision-making in human-agent teams: A pattern approach. In: International Conference on Human-Computer Interaction. pp. 203–220. Springer (2020)

Shepherd: Reinforcement Learning as a Service with Distributed Execution

Hélène Plisnier, Denis Steckelmacher, and Ann Nowé

Vrije Universiteit Brussel (VUB), Brussels, Belgium
dsteckel@ai.vub.ac.be

Video: https://youtu.be/1Ia0MHhhAHg
The web-page: https://steckdenis.be/shepherd_demo.html
Hardware requirements: a power plug and a table on which to put a screen

Abstract. Most current implementations of Reinforcement Learning agents consider that one agent interacts with one environment, and that the agent and environment run on the same machines. Previous work, such as RL-Glue[1], went a step in the direction of allowing the agent and environment to be different processes on a computer, but a wider separation of the agent and environment is much less common. In this demonstration, we illustrate how Shepherd, a web-service that allows clients to remotely query a Reinforcement Learning agent for actions, allows *multiple people* to interact at the same time with a *single agent*, on their phone, over the Internet, without having to install anything. Shepherd ensures that knowledge obtained from one client (one person in this demonstration) is quickly leveraged to improve the performance of the agent for the other clients.

1 The demonstration

This demonstration considers the Reinforcement Learning setting. An agent learns what action to perform in what state of the environment, in order to obtain the highest-possible sum of rewards over an episode (a sequence of actions).

In this demonstration, the environment is the BNAIC venue. We will place Belgian chocolates somewhere on the demo floor, along with a few paper tags. The goal will be for visitors of BNAIC to find the chocolates, by following instructions given by their phones, more precisely, given by a web-page[2] that they have opened on their phone. Every time the visitor finds a tag, they enter on the web-page the two-letters code displayed on the tag, and press a button. The web-page will send a request to a remote Shepherd server, telling it what tag has been seen by the person. The Shepherd server will reply with an instruction, such as "go to the nearest coffee machine" or "look at the demo stand on the right". The user then performs the action. This can lead to 3 outcomes, that the user tells Shepherd about by clicking buttons on the web-page:

[1] https://sites.google.com/a/rl-community.org/rl-glue/Home
[2] https://steckdenis.be/shepherd_demo.html

1. The user finds another tag → they enter its two-letters code and get a new action;
2. The user does not find any tag → they click on a button that *punishes* the Shepherd agent;
3. The user finds the pralines → they click on a button that *rewards* the Shepherd agent.

Over time, the Shepherd agent learns to give instructions that quickly lead the users to the pralines. The agent also learns to avoid instructions that lead to no tag. The main property of this demonstration is that *several people* can participate in the demo at the same time. They will all interact with the Shepherd agent independently, and receive their own instructions. Shepherd makes sure that the experiences collected by one person immediately improve the instructions given to the other people. This is a *single agent, multiple executions* setting, comparable to what A3C proposes for compute-efficient Reinforcement Learning [2]. The novelty of Shepherd is that it does not rely on the A3C algorithm, but instead is compatible with any Reinforcement Learning algorithm.

2 Shepherd

Shepherd is a web application, implemented in Python with Django[3]. It acts as a bridge between web clients, that connect to it over the network (using JSON commands sent over HTTP), and state-of-the-art Reinforcement Learning agents available in the Stable-Baselines3 [4]. Shepherd presents itself to the RL algorithms as a fully standard OpenAI Gym environment [1]. At the core of Shepherd, the Actor-Advisor [3] is used to allow each client of Shepherd, each running their own instance of the RL algorithm, to advice the other clients. This is what allows Shepherd to be compatible with the *single agent, multiple executions* setting described above, without having to modify the RL algorithms it exposes to the clients.

For this demonstration, a Shepherd instance will run on a server visible on the Internet. The Shepherd agent will be configured to learn with Tabular BDPI, a tabular (discrete states) version of Bootstrapped Dual Policy Iteration [6] described in this PhD thesis [5]. BDPI has been chosen because it is highly sample-efficient, especially in its tabular version, which is critical for a demonstration during which the agent will learn (instead of a demonstration that shows an already-trained agent).

Acknowledgments

The first author is funded by the Science Foundation of Flanders (FWO, Belgium), grant number 1SA6619N. The second author is supported by the Flemish AI Program.

[3] `https://www.djangoproject.com/`

References

1. Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, 2016.
2. Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, page 10, 2016.
3. Hélène Plisnier, Denis Steckelmacher, Diederik M Roijers, and Ann Nowé. The actor-advisor: Policy gradient with off-policy advice. In *European Workshop on Reinforcement Learning (EWRL)*, 2018.
4. Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3. `https://github.com/DLR-RM/stable-baselines3`, 2019.
5. Denis Steckelmacher. *Model-Free Reinforcement Learning for Real-World Robots*. PhD thesis, Vrije Universiteit Brussel, 2020.
6. Denis Steckelmacher, Helene Plisnier, Diederik Roijers, and Ann Nowe. Sample-efficient model-free reinforcement learning with off-policy critics. In *European Conference on Machine Learning 2019*, pages 19–34, 2020.

Reinforcement Learning-Based Persuasion by a Conversational Agent for Behavior Change

Nele Albers[1], Mark A. Neerincx[1 2], and Willem-Paul Brinkman[1]

[1] Delft University of Technology, Delft, The Netherlands
{n.albers, m.a.neerincx, w.p.brinkman}@tudelft.nl
[2] TNO, Soesterberg, The Netherlands

Keywords: Conversational agent · Persuasion · Reinforcement learning.

1 Introduction

There is certainly some behavior that you want to change. Maybe you want to become more physically active, call your mother more often or snack less when watching TV at night. Let's assume that you want to quit smoking. You are not doing this alone, but are supported by your coach Hannah. Hannah constantly persuades you to stick to your intervention. How does she decide how to do that? First, Hannah has a lot of theoretical expertise. Moreover, you are not Hannah's first client, so she can draw upon her experience with other and especially similar clients. Third, Hannah considers your current situation - are you confident or stressed about a deadline? In addition, she will persuade you in such a way that she can persuade you again in the future. And lastly, Hannah will keep adapting her strategy over time. Now, let's suppose that you have another coach, Sam. Unlike Hannah, Sam is a virtual coach. Can Sam do what Hannah can?

Changing personal behavior is a very promising way to improve health and reduce premature death. For example, nearly 40% of deaths in the United States are caused by behavior [21][26], and smoking alone contributes to 19,000 annual deaths in the Netherlands [22][29]. To support such behavior change, recent years have seen a surge of eHealth applications [4][8][17][18]. Yet, while such interventions have the advantage that they are available at all times, scalable, cost-effective and can facilitate tailoring [16], adherence to them remains low [4][15]. We thus aim at developing persuasive communication for a virtual coach that aids people in adhering to their intervention. Previous work has shown that data gathered on other people [13][14], similar people [11][30] or an individual [12][13][14][20][25] can be used to choose a persuasion type. Yet, little work has also incorporated the context of a persuasive attempt, which has been supposed to have an important impact on the effectiveness of different persuasion types [2][3][24]. In addition, persuasion types also differ in their impact on the context of future persuasive attempts [28]. We thus propose a reinforcement learning approach to persuading people that considers a person's current and future states as well as the similarity of people. We test this approach based on persuading people to do small preparatory activities for smoking cessation and physical activity increase such as listing reasons for wanting to quit smoking.

2 Approach

We created a text-based virtual coach that attempts to persuade people to do small activities. For each persuasive attempt, the virtual coach selects a persuasion type based on its learned policy. After a certain time interval, the user provides the virtual coach with feedback by reporting the effort they put into their suggested activity. This feedback is used by the agent to update its policy. Formally, we can define our approach as a Markov Decision Process $\langle S, A, R, T, \gamma \rangle$. The action space A thereby consists of different persuasion types, which include a subset of Cialdini's persuasion types [6], action planning [5][10][27], and the option to not persuade. The reward function $R : S \times A \rightarrow [-1, 1]$ is determined by the self-reported effort, $T : S \times A \times S \rightarrow [0, 1]$ describes the transition function, and the discount factor γ is set to 0.85 to favor rewards obtained in the near future over rewards obtained in the more distant future. The finite state space S is defined by answers to questions that are based on the COM-B Model for Behavior Change [19] and capture a person's capability, opportunity and motivation to perform an activity (e.g. "I feel that I need to do the activity"). To further incorporate the similarity of people, the agent maintains a policy π_i for each user i. When updating π_i, an observed sample from user j is weighted based on how similar i and j are with regards to their personality [9] and stage of change for becoming more physically active based on [23].

3 Experiment

To gather data for and test our approach, we have conducted an experiment with more than 500 daily smokers who planned or contemplated to quit smoking [7]. Participants interacted with the virtual coach Sam in five conversational sessions. In each session, the virtual coach suggested a new activity, together with a persuasion type. The first two sessions thereby served as training sessions in which participants were persuaded by a random persuasion type, whereas the last three sessions were used to test the algorithm components. To this end, participants were randomly split into four groups after session 2. Based on the data gathered in sessions 1 and 2, participants in the four groups were subsequently persuaded based on 1) a persuasion type with the highest immediate reward average, 2) a persuasion type with the highest immediate reward average in their state, 3) a persuasion type with the highest Q-value in their state, and 4) a persuasion type with the highest similarity-weighted Q-value in their state. The data from the experiment will be analyzed according to our Open Science Framework (OSF) pre-registration [1]. We will also share our collected data in anonymized form.

Acknowledgments. This work is part of the multidisciplinary research project Perfect Fit, which is supported by several funders organized by the Netherlands Organization for Scientific Research (NWO), program Commit2Data - Big Data & Health (project number 628.011.211).

References

1. Albers, N., Brinkman, W.P.: Perfect fit - experiment to gather data for and test a reinforcement learning-approach for motivating people (May 2021). https://doi.org/10.17605/OSF.IO/K2UAC, osf.io/k2uac
2. Alslaity, A., Tran, T.: On the impact of the application domain on users' susceptibility to the six weapons of influence. In: International Conference on Persuasive Technology. pp. 3–15. Springer (2020). https://doi.org/10.1007/978-3-030-45712-9_1
3. Bertolotti, M., Carfora, V., Catellani, P.: Different frames to reduce red meat intake: The moderating role of self-efficacy. Health Communication **35**, 475 – 482 (2019)
4. Beun, R.J., Brinkman, W.P., Fitrianie, S., Griffioen-Both, F., Horsch, C., Lancee, J., Spruit, S.: Improving adherence in automated e-coaching - A case from insomnia therapy. In: International Conference on Persuasive Technology. pp. 276–287. Springer (2016). https://doi.org/10.1007/978-3-319-31510-2_24
5. Chapman, J., Armitage, C.J., Norman, P.: Comparing implementation intention interventions in relation to young adults' intake of fruit and vegetables. Psychology and Health **24**(3), 317–332 (2009)
6. Cialdini, R.B.: Influence: the psychology of persuasion, revised edition. New York: William Morrow (2006)
7. DiClemente, C.C., Prochaska, J.O., Fairhurst, S.K., Velicer, W.F., Velasquez, M.M., Rossi, J.S.: The process of smoking cessation: an analysis of precontemplation, contemplation, and preparation stages of change. Journal of consulting and clinical psychology **59**(2), 295 (1991)
8. Fadhil, A., Gabrielli, S.: Addressing challenges in promoting healthy lifestyles: the al-chatbot approach. In: Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare. pp. 261–265. ACM (2017). https://doi.org/10.1145/3154862.3154914
9. Gosling, S., Rentfrow, P., Swann, W.: A very brief measure of the big-five personality domains. Journal of Research in Personality **37**, 504–528 (2003)
10. Hagger, M.S., Luszczynska, A.: Implementation intention and action planning interventions in health contexts: State of the research and proposals for the way forward. Applied Psychology: Health and Well-Being **6**(1), 1–47 (2014)
11. Hors-Fraile, S., Malwade, S., Luna-Perejon, F., Amaya, C., Civit, A., Schneider, F., Bamidis, P., Syed-Abdul, S., Li, Y.C., De Vries, H.: Opening the black box: Explaining the process of basing a health recommender system on the i-change behavioral change model. IEEE Access **7**, 176525–176540 (2019). https://doi.org/10.1109/ACCESS.2019.2957696
12. Kang, Y., Tan, A., Miao, C.: An adaptive computational model for personalized persuasion. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI. pp. 61–67. AAAI Press (2015)
13. Kaptein, M., Markopoulos, P., De Ruyter, B., Aarts, E.: Personalizing persuasive technologies: Explicit and implicit personalization using persuasion profiles. International Journal of Human-Computer Studies **77**, 38–51 (2015). https://doi.org/10.1016/j.ijhcs.2015.01.004
14. Kaptein, M., McFarland, R., Parvinen, P.: Automated adaptive selling. European Journal of Marketing **52** (2018)
15. Kelders, S.M., Van Zyl, L.E., Ludden, G.D.: The concept and components of engagement in different domains applied to ehealth: a systematic scoping review. Frontiers in psychology **11**, 926 (2020)

16. Liao, Y., Wu, Q., Tang, J., Zhang, F., Wang, X., Qi, C., He, H., Long, J., Kelly, B.C., Cohen, J.: The efficacy of mobile phone-based text message interventions ('happy quit') for smoking cessation in china. BMC Public Health **16**(1), 1–11 (2016)
17. Ly, K.H., Ly, A.M., Andersson, G.: A fully automated conversational agent for promoting mental well-being: a pilot rct using mixed methods. Internet interventions **10**, 39–46 (2017)
18. Meijer, E., Korst, J.S., Oosting, K.G., Heemskerk, E., Hermsen, S., Willemsen, M.C., van den Putte, B., Chavannes, N.H., Brown, J.: "at least someone thinks i'm doing well": a real-world evaluation of the quit-smoking app stopcoach for lower socio-economic status smokers. Addiction science & clinical practice **16**(1), 1–14 (2021)
19. Michie, S., Atkins, L., West, R., et al.: The behaviour change wheel. A guide to designing interventions. 1st ed. Great Britain: Silverback Publishing (2014)
20. Mintz, Y., Aswani, A., Kaminsky, P., Flowers, E., Fukuoka, Y.: Nonstationary bandits with habituation and recovery dynamics. Operations Research **68**(5), 1493–1516 (2020)
21. Mokdad, A.H., Marks, J.S., Stroup, D.F., Gerberding, J.L.: Actual causes of death in the united states, 2000. Jama **291**(10), 1238–1245 (2004)
22. Nationaal Expertisecentrum Tabaksontmoediging: Kerncijfers roken 2017: De laatste cijfers over roken, stoppen met roken, meeroken en het gebruik van elektronische sigaretten (2018)
23. Norman, G., Benisovich, S., Nigg, C., Rossi, J.: Examining three exercise staging algorithms in two samples. In: 19th annual meeting of the Society of Behavioral Medicine (1998)
24. Oinas-Kukkonen, H., Harjumaa, M.: Persuasive systems design: Key issues, process model, and system features. Communications of the Association for Information Systems **24**(1), 28 (2009)
25. Roy, S., Crick, C., Kieson, E., Abramson, C.: A reinforcement learning model for robots as teachers. In: 27th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN. pp. 294–299. IEEE (2018). https://doi.org/10.1109/ROMAN.2018.8525563
26. Schroeder, S.A.: We can do better—improving the health of the american people. New England Journal of Medicine **357**(12), 1221–1228 (2007)
27. Sniehotta, F.F., Scholz, U., Schwarzer, R., Fuhrmann, B., Kiwus, U., Völler, H.: Long-term effects of two psychological interventions on physical exercise and self-regulation following coronary rehabilitation. International journal of behavioral medicine **12**(4), 244–255 (2005)
28. Steward, W.T., Schneider, T.R., Pizarro, J., Salovey, P.: Need for cognition moderates responses to framed smoking-cessation messages 1. Journal of Applied Social Psychology **33**(12), 2439–2464 (2003)
29. Trimbos Instituut: Richtlijn behandeling van tabaksverslaving en stoppen met roken ondersteuning: Herziening 2016 (2016)
30. de Vries, R.A.J.: Theory-based and tailor-made: Motivational messages for behavior change technology. PhD Thesis (2018)

SafeTraveller - A conversational assistant for BeNeLux travellers*

Kristina Kudryavtseva[1] and Sviatlana Höhn[1][0000−0003−0646−3738]

University of Luxembourg
kristina.kudryavtseva.001@student.uni.lu
sviatlana.hoehn@uni.lu

Abstract. The artificial conversational assistant SafeTraveller helps people understand travel regulations related to COVID-19. The current implementation covers travel regulations for BeNeLux countries. It is implemented using RASA and works in Facebook Messenger. The heuristic-based evaluation of user experience shows performance above average.

Keywords: COVID-19 Travel Regulations · Conversational Assistant

1 Problem

COVID-19 pandemic caused various restrictions in mobility within and across European countries. It lead to a lot of uncertainty in regions close to borders with high number of work commuters. The regulations changed many times, and different rules were applied for transit and stays of different duration in countries, as well as for different travel purposes. People were overwhelmed with changing rules, actual information is sometimes difficult to find, and sometimes only available in one language not spoken by the concerned persons.

While many implementations address issues related to health questions and symptom checking, for example [2,4,3], topics of mobility under pandemic conditions did not receive much attention. We solve these problems with an artificial conversation assistant called SafeTraveller.

2 Solution

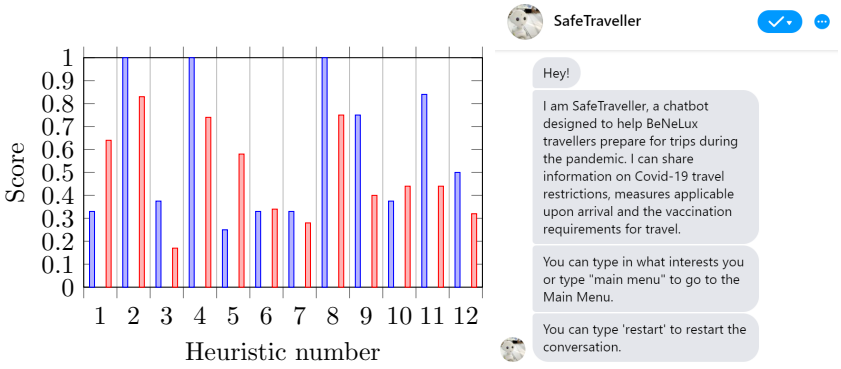
Our first prototype includes travel information within and across three countries: Luxembourg, Belgium and the Netherlands. The chatbot is provided with a knowledge base of travel regulations. The chatbot retrieves an answer depending on the travel characteristics: transit or stay, duration of stay (e.g. more or less than 48 h), purpose of stay (e.g. work or leisure). The chatbot also provides information about COVID-19 tests and informs about wearing masks. It also covers regulations related to vaccinations. The current implementation is based

* S. Höhn thanks Luxembourgish National Research Fund INTER-SLANT 13320890

on RASA[1] and uses Facebook Messenger to connect with users. The knowledge based was populated with regulations related to travel and restrictions within the countries. The regulation texts were paraphrased to make them sound more conversational and accessible. The working of the system is illustrated in a video demonstration of the prototype[2].

3 Evaluation

The evaluation of the user experience in expert interviews based on 12 heuristics [3] shows that SafeTraveller outperforms the average results from [3] in heuristics 2,3,4,8,9,11,12, reaches approximately the same score in heuristics 6,7,10 and needs more attention in heuristics 1 and 5 (SafeTraveller in blue and the average bot in red in the plot below). The screenshot shows the start of the conversation.



4 Conclusions and Future Work

SafeTraveller as a proof-of-concept shows that the conversational assistants help to find the relevant information for the given use case faster. The dynamics of the pandemic (vaccinations, virus mutations and people's mobility) shows that this topic is still urgent.

In our next release of the SafeTraveller we plan to include information for all European countries in at least three languages. We plan to use the dataset collected by [1] to train the language understanding models in all languages of the Greater Region. In addition, we plan to include dynamic updates of the knowledge base in order to keep the information for all countries up to date. We will also integrate logic and reasoning to handle contradictions. However, several research challenges need to be solved such as automated translation of regulations and automated reasoning over a multilingual knowledge base.

[1] https://rasa.com
[2] https://youtu.be/BKuH7lMw3PU

References

1. Chen, N., Zhong, Z., Pang, J.: An exploratory study of covid-19 information on twitter in the greater region. Big Data and Cognitive Computing **5**(1), 5 (2021)
2. Espinoza, J., Crown, K., Kulkarni, O.: A guide to chatbots for COVID-19 screening at pediatric health care facilities. JMIR public health and surveillance **6**(2) (2020)
3. Höhn, S., Bongard-Blanchy, K.: Heuristic evaluation of COVID-19 chatbots. In: Proceedings of CONVERSATIONS. pp. 131–144. Springer (2020)
4. Munsch, N., Martin, A., Gruarin, S., Nateqi, J., Abdarahmane, I., Weingartner-Ortner, R., Knapp, B.: Diagnostic accuracy of web-based COVID-19 symptom checkers: comparison study. Journal of medical Internet research **22**(10) (2020)

Logical Reasoning application with NLP interface to construct the Knowledge Base

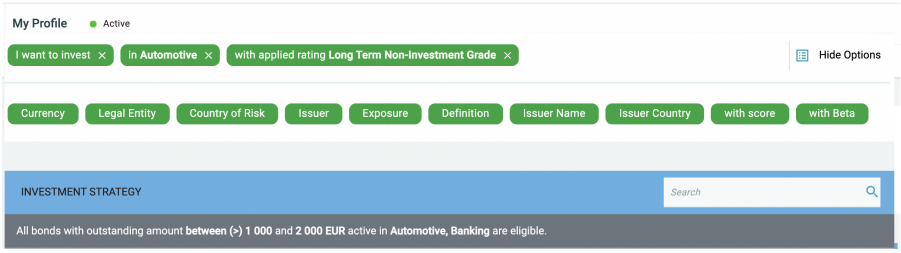
Marjolein Deryck[1,2], Nuno Comenda[3], Bart Coppens[3], and Joost Vennekens[1,2]

[1]KU Leuven, Dept. Computer Science, Campus De Nayer
[2]Leuven.AI – KU Leuven Institute for AI, Leuven, Belgium
[3]Coppens and Partners Consulting
{marjolein.deryck,joost.vennekens}@kuleuven.be
{nuno.comenda, bart.jan.coppens}@coppens-and-partners.com

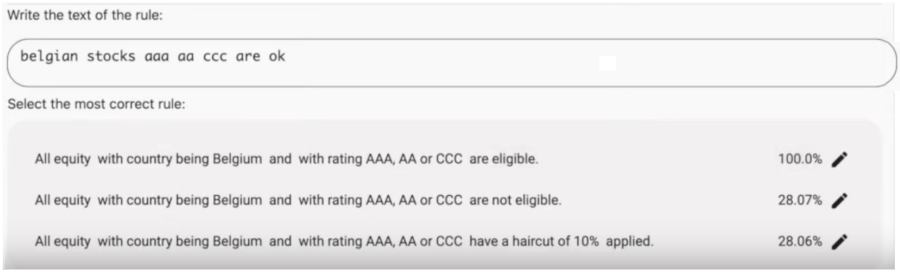
The field of knowledge representation and reasoning (KRR) is under development since the early days of artificial intelligence [6]. One of the largest challenges in the field is to develop a representation that is expressive enough to model a certain domain, but that is also useful to reason with and derive conclusions. The Knowledge Base Paradigm (KBP) advocates a strict separation between these two concerns: domain knowledge should be modelled in a Knowledge Base (KB), while separate inference algorithms can use this KB to calculate solutions for a given problem. The IDP system offers an implementation of the KBP [3]. We have demonstrated the power of the IDP system in a variety of case studies including law, configuration and insurance [5,1,2]. In this demo we showcase a system that consists of a Natural Language (NL) interface that allows an easy creation of the KB, combined with the various functionalities offered by IDP.

The KB in IDP is modelled using FO(·), a language derived of First Order Logic [7]. Although the language is easily readable for computer scientists, this is far from trivial for business users. However, in many applications it is important that business users can adapt the KB themselves. Our contribution is that we created an application that combines KRR with NL processing, such that a natural language profile is automatically translated into a knowledge base that can be used for automated reasoning. The setup of the system is generic, such that it can easily be tweaked to different sectors. E.g., we have successfully developed a production application in the domain of investment management [4]. In this application clients are able to create their own investment profile based on asset characteristics and their associated Environmental, Social en Corporate Governance (ESG) metrics (hence implementing the company's ESG policy in the investment strategy). But we have also developed prototypes for other areas, both within and outside the financial sector, e.g., credit risk applications and client communication templates.

The key element of the natural language processing functionality is the manually created *tuple tree*. The standard tree of a case study application to create investment profiles consists of 400 nodes and took two weeks to construct. The creation time of a tuple tree depends on the number of nodes within the domain. Each node represents a concept, like country or asset type, and each concept accepts possible values related to it. Users can extend the available picklist of values by introducing and defining new concepts. The concepts are used to cre-



(a) CNL building blocks



(b) Natural language sentences

Fig. 1: Adding formulas to the KB using CNL and NLP

ate highly structured sentences step-by-step by selecting blocks (panel(a) of Fig. 1). To avoid ambiguity, the sentences sum up the conditions that need to be true in conjunction. It is not possible to create a sentence with disjunction. Any NL sentence that uses an 'or' statement, can be split in different sentences that only contain conjunctions. The resulting highly structured natural language sentence is automatically translated to FO(·) and added to the KB. The user can also type a free formal natural language sentence (panel(b) of Fig. 1). The application then proposes three CNL statements that are most likely to present the English sentence. The user then selects the most correct sentence, makes adjustments if necessary, and validates the result. As before, this CNL statement is added to the KB in FO(·). The NLP module consists of a custom attention-based network, that was implemented in Tensorflow. The models use a sequence-to-sequence architecture with attention. The training data uses a combination of real and synthetic data. The real data are previous interactions from the users. The synthetic data is achieved by creating random walks on the tuple tree and several grammatical transformations to achieve a rich set of examples. The model is trained with a combination of real and synthetic data on an NVIDIA RTX 2080 with training times from tree to five hours.

Once the KB is created, numerous powerful and generic inference tasks can be used and combined such as model expansion, optimisation, propagation, explanation, etc. [7]. As a result, the system can offer multiple services that i) use the same KB, and ii) are inconceivable in classic imperative system.

Acknowledgements

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

1. Bram Aerts and Joost Vennekens. An application of logic-based methods to machine component design. volume 64, pages 13:1–13:15. Palù, Alessandro Dal, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2018.
2. Ingmar Dasseville, Laurent Janssens, Gerda Janssens, Jan Vanthienen, and Marc Denecker. Combining {DMN} and the knowledge base paradigm for flexible decision enactment. In *RuleML 2016 Supplementary Proceedings*, 2016.
3. Broes De Cat, Bart Bogaerts, M Bruynooghe, G Janssens, and Marc Denecker. Predicate logic as a modeling language: The idp system. In *Declarative Logic Programming: Theory, Systems, and Applications*, pages 279–329. ACM Books, 2018.
4. Marjolein Deryck, Nuno Comenda, Bart Jan Coppens, and Joost Vennekens. Combining Logic and Natural Language Processing to Support Investment Management. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, page 5, Hanoi, 2021.
5. Marjolein Deryck, Jo Devriendt, Simon Marynissen, and Joost Vennekens. Legislation in the knowledge base paradigm: interactive decision enactment for registration duties. pages 174–177. IEEE, 2019.
6. John Haugeland. *Artificial Intelligence: The Very Idea*. Massachusetts Institute of Technology, USA, 1985.
7. Johan Wittocx, Maarten Mariën, and Marc Denecker. The idp system: a model expansion system for an extension of classical logic. In *Proceedings of the 2nd Workshop on Logic and Search*, pages 153–165. ACCO; Leuven, 2008.

Using privacy preserving amalgamated machine learning for pedestrian safety in warehouses

Imen Chakroun, Tom Vander Aa, Roel Wuyts and Wilfried Verarcht

Exascience Life Lab , IMEC
Leuven , Belgium

Forklifts and automated guided vehicles (AGV) are useful tools in factories and distribution facilities. With regard to worker safety, however, there are issues to keep in mind when using either manned or automated powered industrial vehicles. Recent information about automated guided vehicle accidents also demonstrates that even with on-board sensors these vehicles did not detect nearby workers. In many forklift incidents, the driver's view was partially or fully blocked due to the forklift structure and load, environment occlusions, etc. Further complications involving worker safety will happen as AGVs work in more unstructured environments. These safety risks can be mitigated with the use of new machine learning techniques that run models using on-board and environment sensors.

Consider a setup (Figure 1) where multiple cameras view (different parts of) a scene and need to give environmental feedback to one or more AGVs that risk an imminent collapse because of occlusion. We see every camera as a separate privacy silo that does not share its raw data, nor its internal models as every device have only a partial view on the global image.

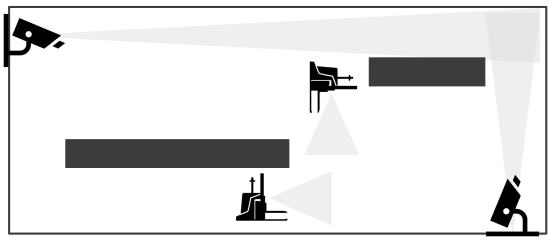


Fig. 1. Sketched scenario with AGVs and environment cameras.

One solution would be to pool all data from all cameras (stitching frames together or centralizing data) and learn on that combined dataset (Figure 2). However this may be problematic for technical reasons (required network bandwidth, centralized storage) or for privacy concerns. To tackle this, We have defined a privacy preserving amalgamated machine learning (PPAML) solution (Figure 3). Our amalgamated machine learning technique lets every local model build intermediate features (dubbed PAML features) that can be safely shared with other models. The camera has its own model that calculates PAML features that

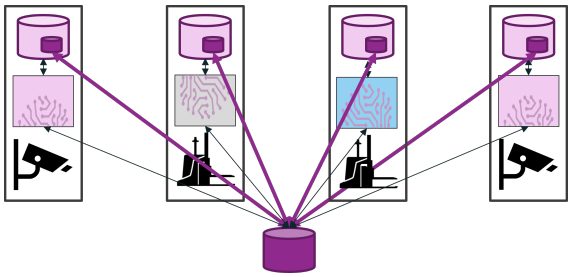


Fig. 2. Pooling all data together.

are used in the AI model driving the AGV. Even though every device knows only its portion of the network and its data, we showed experimentally that overall prediction is improved with respect to the local models.

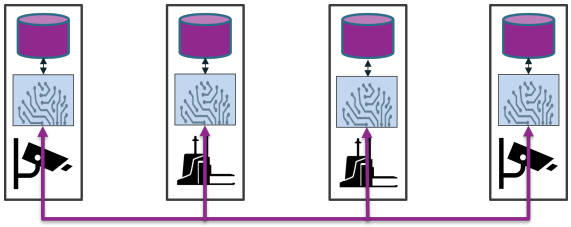


Fig. 3. Using PPAML features.

We have build a physical demonstrator for pedestrian safety in warehouses where pedestrian and automated guided vehicles (AGV) work together. Using PPAML, the environment sensor detects pedestrian and AGV positions and sends PPAML features to the AGV which uses these abstract features in an internal machine learning model to quantify the safety of the environment. The AGV do not have any sensor on-board. It relies entirely on the input of the camera. For the communication layer between the environment sensor and the robot we are leveraging the DUST framework [1]. When the robot predicts that it is very close to the pedestrian it will step back immediately to further proceed when it predicts safety again. We currently have implemented a solution with a single robot and environment sensor, which we then plan to extend further in the future. The demonstration can be found in https://www.dropbox.com/s/rtkq2zlmljqd31y/DemoChakrounetal.mp4?dl=0

1 Acknowledgement

We thank the researchers from IDLab Antwerp to get us up and running with the DUST framework.

References

1. Vanneste, Simon and de Hoog, Jens and Huybrechts, Thomas and Bosmans, Stig and Eyckerman, Reinout and Sharif, Muddsair and Mercelis, Siegfried and Hellinckx, Peter. Distributed Uniform Streaming Framework: An Elastic Fog Computing Platform for Event Stream Processing and Platform Transparency. Future Internet. Volume 11. 2019.

A Machine Translation powered AI Chatbot

Dimitra Anastasiou[1], Anders Ruge[2], Hoorieh Afkari[1], Patrick Gratz[1], Radu Ion[3], Verginica Barbu Mititelu[3], Olivier Pedretti[1], Svetlana Segarceanu[4], George Suciu[4]

[1] Luxembourg Institute of Science and Technology
[2] SupWiz Aps, Denmark
[3] Research Institute for Artificial Intelligence, Romanian Academy
[4] BEIA Consult International, Romania
dimitra.anastasiou@list.lu

Abstract. ENRICH4ALL is a newly funded project about the development of an e-government chatbot. The Machine Translation system of the European Commission, *eTranslation*, will be integrated into a commercial AI chatbot engine and through fine-tuned Natural Language Understanding models, a newly developed multilingual chatbot service will be deployed in Luxembourg, Denmark, and Romania.

Keywords: AI, eTranslation, Multilingual Chatbot, NLU.

1 Introduction

1.1 ENRICH4ALL project

In this paper we introduce the European Action ENRICH4ALL [1] (E-goverNment [RI] CHatbot for ALL) which is about the development of a multilingual chatbot service called *eChat* that will be deployed in public administration in Denmark, Luxembourg, and Romania.

Government chatbot. The COVID pandemic has drastically changed how governmental services have been working so far and how Digital Service Infrastructures (DSIs) deliver networked cross-border services for citizens, businesses, and public administrations. The "digital first" mindset plays a big role in today's society and tends to be the next new normal. The advancements in AI and Machine Learning have made virtual assistants very powerful and present in many domains nowadays, such as commerce, healthcare, etc. Microsoft stated that the ultimate form of AI is a digital assistant and in 20 years, AI-operated personal digital assistants will be so integrated into our lives that they will be like "alter egos, a second self" [2]. Regarding government chatbots in Europe, there is not currently any interoperable infrastructure throughout the multiple EU countries. There are a few existing e-Government chatbots, such as in Estonia [3], Belgium [4], Ukraine [5], and UK [6], but they do not share the same architecture, and thus are not interoperable.

With *eChat*, we aim to integrate *eTranslation* [7], and move from a scarce and fragmented e-government virtual assistant-based interaction to a fully digital and unified

ecosystem, which provides updated information on laws, regulations and public services 24/7. E-government chatbots have many benefits for all stakeholders: citizen, businesses and public administration. Chatbots can process many service requests, work 24/7, and provide always timely and up-to-date information. On the top of that, having a multilingual government chatbot, the services are open to a large number of people irrespective of language, geographical or cultural barriers, see Sec. 2.

In ENRICH4ALL we will use the AI powered chatbot by SupWiz, Denmark [8] and integrate the eTranslation API (see 2.1). The SupWiz chatbot is based on Natural Language Understanding (NLU) models which will be fine-tuned to appropriately map the user questions, so called *intents* to the chat flows that will be developed in the project. Besides, the SupWiz chatbot is easy to set-up, enables smooth integrations, and also assists human agents by transferring the intents to a department, when the user asks for human support or the chatbot cannot handle the user´s problem.

2 Multilingual chatbot

2.1 eTranslation powered AI chatbot

eTranslation is the neural Machine Translation (MT) tool provided by the European Commission to all EU bodies, public services, and public administrations across EU, Iceland and Norway, as well as European SMEs and startups. It currently covers not only the 24 official languages of the EU, but also Russian and simplified Chinese, Turkish, and Arabic. *eTranslation* is a Connecting Europe Facility (CEF) building block which can be integrated into digital services in order to add translation capabilities. Thus, *eTranslation* is available both as stand-alone webservice and as API that can be integrated in other online services. One significant benefit of *eTranslation* over other MT solutions for a government chatbot solution implementation is the data privacy preservation. This privacy will be even enhanced through a user profiling module that will be developed in ENRICH4ALL: user data will be extracted and identified in order to have the history logs from previous user-agent interactions; this user data is limited to the name and a unique ID number. The history logs will enable the facilitation and acceleration of personalized user requests.

2.2 Societal impact

Luxembourg is a highly multilingual country with 20% of the population speaking three languages at work environment [9]. Moreover, Luxembourgish is not yet an official EU language. However, Luxembourgish is spoken by the majority of the population as a 1st and 2nd language and thus, a multilingual chatbot in Luxembourg, supporting also Luxembourgish, would have a significant economic and societal impact. ENRICH4ALL is in line both with Luxembourg's strategic vision for AI [10] as well as with the resolution "Language equality in the digital age", which was passed by the European Parliament in 2018. Motivated by this resolution, the European Language Equality [11] project, consisting of 52 partners covering all European countries,

research and industry and all major pan-European initiatives, develops a strategic research, innovation and implementation agenda as well as a roadmap for achieving full digital language equality in Europe by 2030. Through a multilingual bot, citizen would save their long way to the public administration and would not be hindered by any language barrier, since they can communicate with the *eChat* in any of their preferred language. More significantly, the public administrations that will deploy the chatbot will reduce their resources having a bot to resolve easy issues, such as password resets or getting information on laws and regulations, while the human agents can have the possibility to focus on more complex requests, which cannot be resolved yet digitally.

Acknowledgment

The Action 2020-EU-IA-0088 has received funding from the European Union's Connecting Europe Facility 2014-2020 - CEF Telecom, under Grant Agreement No. INEA/CEF/ICT/A2020/2278547.

References

1. ENRICH4ALL Homepage, https://www.enrich4all.eu/, last accessed 2021/08/24.
2. Smith, Brad, and Harry Shum. "The Future Computed." *Artificial Intelligence and its role in society* (2018).
3. https://e-estonia.com/ai-chatbot-to-replace-and-improve-governmental-e-services/, 2021/10/13.
4. https://northsearegion.eu/media/12352/used-case-chatbot.pdf, 2021/10/13.
5. Petriv, Y., Erlenheim, R., Tsap, V., Pappel, I., & Draheim, D. (2019). Designing effective chatbot solutions for the public sector: A case study from Ukraine. *International Conference on Electronic Governance and Open Society: Challenges in Eurasia*, 320-335, Springer, Cham.
6. https://tfl.gov.uk/info-for/media/press-releases/2017/june/tfl-launches-new-social-media-travelb, 2021/10/13.
7. https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eTranslation, 2021/10/13.
8. https://www.supwiz.com/chatbot-1, last accessed 2021/08/27.
9. https://digital-luxembourg.public.lu/stories/luxembourgs-strategic-vision-ai, last accessed 2021/08/30.
10. https://statistiques.public.lu/catalogue-publications/regards/2019/PDF-09-2019.pdf
11. https://european-language-equality.eu/, 2021/10/13.

Talking to your Data: Interactive and interpretable data mining through a conversational agent*

Isel Grau[1][0000−0002−8035−2887], Luis Daniel Hernandez[1], Astrid Sierens[1], Simeon Michel[2], Nico Sergeyssels[2], Vicky Froyen[3][0000−0002−5649−5888], Catherine Middag[2][0000−0001−5732−0281], and Ann Nowe[1][0000−0001−6346−4564]

[1] Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium
[2] Gezondheidszorg, Design & Technologie, Erasmushogeschool Brussel, Belgium
[3] Collibra NV, Belgium

Abstract. In this demo, we showcase the "Talking to your Data" system. The key idea of this system is to support data governance and data mining in a novel way. We aim to bring the use of interpretable machine learning techniques closer to the business analysts by using natural language. We have developed a conversational agent and a data mining backend that supports the analysis of data. Our approach facilitates solving prediction tasks and also provides explanations for these predictions. Furthermore, we make possible the interaction for including the feedback of the business analysts in the models.

Keywords: conversational agents · decision tree · subgroup discovery · interactive machine learning · explainable artificial intelligence

1 Introduction

The Collibra project [6] aims to develop a platform for supporting data management through smart engagement using a conversational agent. The goal is to go beyond the level of reports, incorporating interpretable data mining models to gain new insights into the data. Here, by interpretability we refer to the transparency of the model, i.e. the model is referring to terms familiar to the user and the user can understand the reasoning of the model [4].

By adding the human in the process of building or fine-tuning a machine learning model, the users' understanding and trust of the system, as well as the accuracy of learned systems, can be improved. Decisions trees (DT) [8] are one of the most widely used intrinsically interpretable machine learning techniques [7]. While the lesser-known (but also interpretable) subgroup discovery techniques are focused on generating descriptions of interesting patterns in data [5]. Other works have proposed interactive machine learning tools for building and visualizing machine learning models [10, 9], but they rely on traditional graphical user interfaces.

* Supported by the Innoviris TeamUp project "Driving collective data governance through smart engagement platforms".

2 System description

In this work, we propose the use of a conversational agent for the interaction with data and interpretable machine learning techniques. The conversational agent was implemented using the RASA library [1], which facilitates the dialogue management and language understanding/generation modules. Our systems relies in two backend services, the Collibra Data Governance Platform [2] and a data mining backend. The Collibra Platform manages all requests regarding reports, data editions, and permissions, while the data mining backend processes all machine learning-related tasks, such as learning, prediction, interpretation, and edition of the models. The system architecture is depicted in Figure 1.

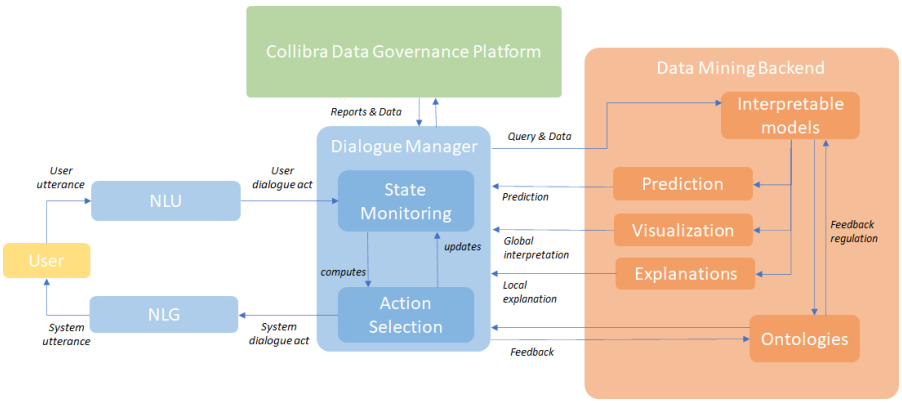


Fig. 1. System architecture of "Talking to your Data", involving the conversational agent (blue), the Collibra Platform backend (green) and the data mining backend (orange).

Our system supports common operations with data that are needed during the exploratory phase of the data mining process. For example, loading or merging datasets, requesting the possible values of a feature, and performing group-by operations offering aggregation statistics. For the predictive phase, we allow to train decision tree models and subgroup discovery algorithms. The latter also providing the possibility of intervening during the optimization process [3]. After the machine model is built, it can be questioned in natural language for obtaining predictions, even with incomplete information. Perhaps the most relevant features are the possibility to obtain explanations over the predictions in the form of rules and to modify those rules based on the feedback of the user, thus changing the trained model with the knowledge of the expert. For this last feature, we rely on ontologies associated with the datasets, which allows controlling the vocabulary, finding alternative features, and reusing the calculations already performed by the classifier. *System requirements for demonstration: Two screens and internet connection.* Video available at: https://youtu.be/SaigB3usp6U

References

1. Bocklisch, T., Faulkner, J., Pawlowski, N., Nichol, A.: Rasa: Open Source Language Understanding and Dialogue Management (2017), http://arxiv.org/abs/1712.05181
2. Collibra: Collibra Data Governance: Organize and understand your data — Collibra, https://www.collibra.com/data-governance
3. Dzyuba, V., van Leeuwen, M.: Interactive discovery of interesting subgroup sets. In: International Symposium on Intelligent Data Analysis. pp. 150–161. Springer (2013)
4. Grau, I., Sengupta, D., Garcia Lorenzo, M.M., Nowé, A.: An Interpretable Semi-supervised Classifier using Rough Sets for Amended Self-labeling. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE (2020)
5. Helal, S.: Subgroup Discovery Algorithms: A Survey and Empirical Evaluation. Journal of Computer Science and Technology **31**(3), 561–576 (2016). https://doi.org/10.1007/s11390-016-1647-1
6. Loeckx, J., Grau, I., Sergeyssels, N., Michel, S., Froyen, V., Middag, C., Nowe, A.: Driving Collective Data Governance through Smart Engagements Platforms (Collibra) (2018)
7. Molnar, C.: Interpretable Machine Learning. Leanpub (2019)
8. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
9. Van Den Elzen, S., Van Wijk, J.J.: BaobabView: Interactive construction and analysis of decision trees. In: VAST 2011 - IEEE Conference on Visual Analytics Science and Technology 2011, Proceedings. pp. 151–160 (2011). https://doi.org/10.1109/VAST.2011.6102453
10. Ware, M., Frank, E., Holmes, G., Hall, M., Witten, I.H.: Interactive machine learning: Letting users build classifiers. International Journal of Human Computer Studies **55**(3), 281–292 (2001). https://doi.org/10.1006/ijhc.2001.0499

An invariants based architecture for combining small and large data sets in neural networks

Roelant Ossewaarde[1 [0000−0002−7036−522X]], Stefan Leijnen[1 [0000−0002−4411−649X]], and Thijs van den Berg[1 [0000−0003−2561−0537]]

Artificial Intelligence Research Group, HU University of Applied Science, Utrecht, The Netherlands
roelant.ossewaarde@hu.nl
stefan.leijnen@hu.nl
thijs.vandenberg@student.hu.nl

Abstract. We present a novel architecture for an AI system that allows a priori knowledge to combine with deep learning. In traditional neural networks, all available data is pooled at the input layer. Our alternative neural network is constructed so that partial representations (invariants) are learned in the intermediate layers, which can then be combined with a priori knowledge or with other predictive analyses of the same data. This leads to smaller training datasets due to more efficient learning. In addition, because this architecture allows inclusion of a priori knowledge and interpretable predictive models, the interpretability of the entire system increases while the data can still be used in a black box neural network. Our system makes use of networks of neurons rather than single neurons to enable the representation of approximations (invariants) of the output.

Keywords: Interpretability · Neural Network architecture · A priori knowledge.

1 Introduction

One problem in machine learning is the combination of data sets of different sizes. In many practical applications, there is domain specific information available that could beneficially influence the training of deep learned data sets. Predictive models based on small data sets often have the advantage that white box AI techniques (interpretable), such as Generalized Linear Models, perform as well as black box AI techniques (less interpretable), such as Artificial Neural Networks (ANNs) [2]. The traditional strategy to combine data in deep learning is based on the pooling of data with different levels of detail level into one input set that is used to train the model. Once pooled, the interpretability properties of the original data sets converge to those of the deep learned model.

ANN may combine different kinds of data through skip layers (ResNet, [1]), gate units (LSTM, [4]) or other architectural devices that allow information of different levels of detail to mix in the training stage. The resulting model mostly

disallows of meaningful interpretation because it is notoriously hard to interpret the assigned weights of a neural network in terms of a representation that is understandable by humans.

2 Computational model with invariants

In human brains objects are represented simultaneously at different detail levels [3]. Higher level representations are more abstract, hence less sensitive to variations that appear in the more detailed layers. In a machine learning system for image recognition, invariance to visual translations (such as scaling, rotations) can be built up by simply memorizing examples that underwent such translations. The core of our computational model is to construct a system that does not rely on such extensive memorization, but that can rather build up the different levels of representation directly and apply any necessary translations - or in this case: add the bias that is learned from smaller data sets.

We present a Neural Network architecture that can learn invariants from data while combining with predictive models of other kinds to arrive at a joint prediction that retains both the predictive power and the interpretability properties of each of the individual data models.

The proposed general computational architecture follows a representational model of invariants [3]. To compute invariants, we implement a distinction between *simple* and *complex* neurons (henceforth: S-, respectively C-neurons) where C-neurons pool S-neurons in a network.

3 Impact and limitations

The impact of our architecture on practical implementations is that a priori bias or well-known functions may be mixed into the predictions made by the neural network. This allows prior established bias to occur in the network, which reduces the size of training and enhances its understandability.

One of the notable limitations is that our architecture requires more engineering steps than traditional systems that learn all traits from a single input layer. In addition, the performance of ANNs may be better if the system is allowed to learn from input data in an unconstrained way - unlike in our approach, which uses a priori knowledge to influence the output of the system.

Thesis abstracts

BNAIC/BeneLearn proceedings
November 10–12, 2021
Belval, Esch-sur-Alzette (Luxembourg)

Automated Diagnostic System of Skin Cancer using Deep Convolutional Neural Networks on Dermoscopic Images

Wafaa Aljbawi[1,2] and Alexia Briassouli[1,3]

[1] Maastricht University, Paul-Henri Spaaklaan 1, 6229 EN Maastricht, Netherlands
[2] w.aljbawi@student.maastrichtuniversity.nl
[3] alexia.briassouli@maastrichtuniversity.nl

1 Introduction

Melanoma is considered the deadliest type of skin cancer, due to its potential to spread more easily to other organs, unless it is detected early. Early detection of melanoma is often difficult, with the success rate of diagnosing melanoma by dermatologists reaching 75-84% [2]. The time needed to analyze skin lesions images is one of the many reasons why visual analysis and manual inspection of melanoma are not very reliable [1]. The goal of this paper is to develop a computer-aided diagnostics system that facilitate the early detection of melanoma using both visual and patient information, to assist medical practitioners in diagnosing melanoma effectively. We examine what image preprocessing and data augmentation methods are best suited for combining with State-of-the-Art (SoA) deep learning classification models applied to benchmarking, real-world images of melanoma. We deploy these within a 2 branch deep network, to analyze demonstrate that fusing patient-level contextual information with the analyzed visual data leads to improved melanoma recognition outcomes.

2 Methods

Although research has achieved expert-level performance on skin cancer classification, no study to date has examined patient-level contextual information. To fill the gap between clinical practice and automated melanoma diagnosis, this paper implements a deep learning model that is trained on both skin lesion images and patient-level contextual information. The used patient's metadata consists of age, gender, and anatomical location of skin lesions. We designed a deep neural network by incorporating the patient-level contextual information with lesion images [4]. Specifically, in the proposed model, two branches are designed to handle both the lesion images and clinical information about the patients: The branches are: Image branch which is a pre-trained VGGNet-16 convolutional neural network that responsible for processing image data. Additionally, other models e.g. EfficientNet and ResNet were trained on the same

dataset to compare their performance with VGG-16. The second branch is a basic fully connected neural network that include the patient's information. To begin, the patient's metadata are pre-processed. Then, images are augmented and pre-processed. In the end, the two branches are combined to a single network for the diagnosis of melanoma.

The used dataset contains 10,982 testing images and 33,126 training images of skin lesions from over 2,000 patients (32542 benign and 584 malignant). Considering that there is a significant disproportion among the number of examples of benign and malignant melanoma classes in the dataset, we performed data augmentation techniques to reduce the possible overfitting of the model. Thus, the used data augmentation techniques are: random rotation, image shifting, randomly zooming in the images, random brightness change and shearing.

Pre-processing of the images involves the following steps: body hair augmentation, hair removal because hair may cause a serious information loss, noise reduction, and testing various image features such as low contrast images.

3 Results and Discussion

When the model trained on images with artificial hair, the VGG-16 model obtained 90% accuracy. The accuracy of ResNet50 was lower by 8% compared to VGG-16. But, the model loss was similar to that of the VGG-16 + artificial hair model. By contrast, EfficientNetB0 achieved the best training accuracy of 98%. After data augmentation, training and testing accuracy rose by 17% and 18%, respectively. The model loss was also reduced by 0.031. The findings emphasize the importance of using data augmentation to train melanoma classification models.

We obtained training and testing accuracy of 63% and 64%, respectively, when the images were cropped to focus entirely on the melanocytic lesions. This finding may be explained by the idea that certain essential image features needed to diagnose melanoma can be obscured when images are circularly cropped. Then, converting the images to HSV or Lab color space improved the accuracy greatly. When the model was trained on images in HSV color space, the training and testing accuracy was 96% and 97%, respectively. However, when images were in Lab color space, the model attained an accuracy of 74%. The findings clearly suggest that the color spaces used had a significant impact on the overall model performance.

When the model was trained on only images, the CNN model demonstrated a wide gap between training and testing loss, as well as a poor accuracy of 58%, indicating an overfitting issue. On the contrary, once the model was given extra information about the patient (age and gender), its performance rose significantly from 58% to 90%. In addition, the test loss has been reduced. Surprisingly, when all of the patient's metadata were supplied, model performance fell down. In other words, the accuracy dropped from 90% to 72%. However, this combination of all of patient's metadata and images outperforms training the model with images only.

References

1. Adegun, A., Viriri, S.: Deep learning techniques for skin lesion analysis and melanoma cancer detection: a survey of state-of-the-art. Artificial Intelligence Review **54**(2), 811–841 (2021)
2. Argenziano, G., Soyer, H.P., Chimenti, S., Talamini, R., Corona, R., Sera, F., Binder, M., Cerroni, L., De Rosa, G., Ferrara, G., et al.: Dermoscopy of pigmented skin lesions: results of a consensus meeting via the internet. Journal of the American Academy of Dermatology **48**(5), 679–693 (2003)
3. Siegel, R.L., Miller, K.D., Fuchs, H.E., Jemal, A.: Cancer statistics, 2021. CA: a Cancer Journal for Clinicians **71**(1), 7–33 (2021)
4. Wang, Y., Gong, D., Yang, J., Shi, Q., Hengel, A.v.d., Xie, D., Zeng, B.: An effective two-branch model-based deep network for single image deraining. arXiv preprint arXiv:1905.05404 (2019)

Deepfake Video Detection using Deep Convolutional and Hand-Crafted Facial Features with Long Short-Term Memory Network

Sven van Asseldonk and Itir Önal Ertugrul

Department of CSAI, Tilburg University, Tilburg, The Netherlands
s.j.a.vanasseldonk@tilburguniversity.edu, i.onal@tilburguniversity.edu

Abstract. In this thesis, we propose a novel Deepfake video detection model that extends state-of-the-art spatiotemporal models by adding hand-crafted facial features into the model. The proposed model has been tested against several baseline models on the balanced Celeb-DF dataset with multiple frame selection methods. We conclude that the proposed model outperforms the baseline CNN+LSTM model, but that deep convolutional features are superior to hand-crafted facial features. Finally, this work shows that a frame selection method based on equal intervals captures more inconsistencies, leading to the best performing model. Code for this paper is publicly available at: https://github.com/sjasseldonk/Deepfake-Detection.

Keywords: Deepfake Detection · Hand-Crafted Features · CNN+LSTM.

1 Introduction

Manipulation of visual content that leads to misinformation has become one of the greatest challenges in digital society [11]. Especially facial manipulations are preferred over other objects because faces play a central role in the communication between humans [5]. This phenomenon is known as Deepfakes, stemming from 'Deep Learning' and 'fake', and is defined by [15] as swapping the faces of two persons based on a deep learning approach. As a result of the rapid advancements in Artificial Intelligence (AI) there is an increasing concern that Deepfakes are used for more harmful purposes such as politics [10], causing an erosion of trust [3].

Most Deepfake detection approaches are based on frame-level features present in an image extracted via Convolutional Neural Networks (CNN) [12, 14, 17]. However, these detection models fail to capture information hidden over time in a video, which are referred to as temporal features. Recently, research has been carried out to include these temporal features, outperforming frame-level methods [8, 16, 6]. However, no studies have been found that also include hand-crafted facial features to improve the classification performance, although [13] have found that hand-crafted features may provide models with complementary information. This paper explores the fusion of hand-crafted facial features with deep features and then feeding them to an Long Short-Term Memory (LSTM) network to detect Deepfake videos.

2 Methods

The proposed model in this study fuses deep convolutional and hand-crafted facial features which are then used in a LSTM unit to compose a robust method to detect Deepfake videos. The deep convolutional features are extracted with the Xception architecture [2] pre-trained on the ImageNet dataset [4]. The 84-dimensional hand-crafted facial feature vector include amplitude, velocity and acceleration signals of 17 facial Action Units (AU) [1], eye gaze direction vectors, head pose vectors and distance vectors between two facial landmarks. This hand-crafted feature vector is then concatenated with the deep convolution feature vector derived from the Xception network, composing the 340-dimensional feature descriptor of the model. This feature descriptor is then passed through a single LSTM layer of 512 hidden units to capture temporal inconsistencies. The output of this layer is extended with a 256-dimensional fully connected layer with 50% dropout rate to avoid overfitting. Finally, a 2-dimensional fully connected layer is added with softmax activation function to compute the probabilities of a video being real or Deepfake. Next to the proposed model, we developed 2 baseline models: (i) CNN+LSTM (Baseline 1) and (ii) hand-crafted facial features + LSTM (Baseline 2). We evaluated these models by selecting the first k consecutive frames of the video and by selecting frames with equal intervals.

Table 1. Detection performances on balanced Celeb-DF [7] test set using different frame selection methods. EI refers to a frame selection method with Equal Intervals of 30 and 15 frames in between. The highest performances are marked in bold.

	Frame Selection Method							
	First 10		**First 20**		**EI(30)**		**EI(15)**	
Models	**Acc %**	**AUC**	**Acc %**	**AUC**	**Acc %**	**AUC**	**Acc %**	**AUC**
Baseline 1	0.706	0.775	0.670	0.774	0.731	0.801	0.725	0.819
Baseline 2	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
Proposed	0.681	0.745	0.725	0.801	0.712	0.789	**0.738**	**0.822**

3 Results

The results of this study indicate that hand-crafted facial features can increase the detection accuracy of the model when it has been trained on a sufficient number of frames per video. Besides, deep convolutional features are superior to hand-crafted facial features as the model was not able to learn after removing the CNN extracted features. Although the best performing proposed model did outperform the baseline CNN+LSTM model by 1.3% percent on the Celeb-DF test set, it seems that hand-crafted facial features are becoming less informative features due to the rapid development of sophisticated Deepfake creation methods which is in line with the literature [9]. Lastly, the findings show that selecting 10 frames with an equal interval of 15 frames in between, captures more inconsistencies and irregularities and leads to the best performing model.

References

1. Baltrusaitis, T., Zadeh, A., Lim, Y.C., Morency, L.P.: Openface 2.0: Facial behavior analysis toolkit. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). pp. 59–66. IEEE (2018)
2. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017)
3. Chu, D., Demir, I., Eichensehr, K., Foster, J.G., Green, M.L., Lerman, K., Menczer, F., O'Connor, C., Parson, E., Ruthotto, L., et al.: White paper: Deep fakery - an action plan (2019)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. IEEE (2009)
5. Frith, C.: Role of facial expressions in social interactions. Philosophical Transactions of the Royal Society B: Biological Sciences **364**(1535), 3453–3458 (2009)
6. Güera, D., Delp, E.J.: Deepfake video detection using recurrent neural networks. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6 (2018). https://doi.org/10.1109/AVSS.2018.8639163
7. Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S.: Celeb-df: A large-scale challenging dataset for deepfake forensics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3207–3216 (2020)
8. de Lima, O., Franklin, S., Basu, S., Karwoski, B., George, A.: Deepfake detection using spatiotemporal convolutional networks. arXiv preprint arXiv:2006.14749 (2020)
9. Matern, F., Riess, C., Stamminger, M.: Exploiting visual artifacts to expose deepfakes and face manipulations. In: 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW). pp. 83–92. IEEE (2019)
10. O'Sullivan, D.: Lawmakers warn of 'deepfake' videos ahead of 2020 election (Jan 2019), https://edition.cnn.com/2019/01/28/tech/deepfake-lawmakers/index.html
11. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Niessner, M.: Faceforensics++: Learning to detect manipulated facial images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1–11 (2019)
12. Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., Natarajan, P.: Recurrent convolutional strategies for face manipulation detection in videos. CoRR **abs/1905.00582** (2019), http://arxiv.org/abs/1905.00582
13. Tianyu, Z., Zhenjiang, M., Jianhu, Z.: Combining cnn with hand-crafted features for image classification. In: 2018 14th IEEE International Conference on Signal Processing (ICSP). pp. 554–557 (2018). https://doi.org/10.1109/ICSP.2018.8652428
14. Tolosana, R., Romero-Tapiador, S., Fierrez, J., Vera-Rodriguez, R.: Deepfakes evolution: Analysis of facial regions and fake detection performance. arXiv preprint arXiv:2004.07532 (2020)
15. Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., Ortega-Garcia, J.: Deepfakes and beyond: A survey of face manipulation and fake detection. Information Fusion **64**, 131–148 (2020)
16. Wang, Y., Dantcheva, A.: A video is worth more than 1000 lies. comparing 3dcnn approaches for detecting deepfakes. In: FG'20, 15th IEEE International Conference on Automatic Face and Gesture Recognition, May 18-22, 2020, Buenos Aires, Argentina. (2020)

17. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Two-stream neural networks for tampered face detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1831–1839 (2017). https://doi.org/10.1109/CVPRW.2017.229

Generating common-sense scene graphs using a knowledge base BERT model

C.J.A. Slewe[1], *Supervisors:* dr. M. H.T. de Boer[2], dr. T. Deoskar[1]

[1] Utrecht University, Utrecht, The Netherlands
[2] Netherlands Organisation for Applied Scientific Research (TNO), The Hague, The Netherlands

Keywords: Knowledge Bases, Natural Language Processing, BERT, Scene Graphs.

Introduction. Scene graphs can be used to improve upon autonomous robots by describing a variety of environments [1][2][3]. A scene graph is a symbolic, graphical representation of an image. The nodes correspond to objects in the image, and the edges represent an interaction This thesis compares performance of transformer models, trained on common knowledge bases such as Wikipedia, WordNet [4] and ConceptNet [5], in the creation of common-sense graphs. These graphs are based on images, but the concepts in the graphs are image-independent. For example, a 'living room' can be used as a scene, with a table and a chair present as objects in the common-sense graph. The hypothesis is that the bidirectional encoder representations from transformers (BERT) model can help improve the graph generation by predicting spatial relations between objects [6]. KnowBERT is a version of BERT that uses an entity linker to provide information from a knowledge base to expand the entity embeddings provided to the language model [7]. The combination of knowledge bases and powerful machine learning techniques in KnowBERT make it a well-suited model for spatial relation prediction.

Method. The Visual Genome dataset [8] was used for training. In this thesis, only the triplets with two objects and a relationship were used in order to create a scene graph from a scene as input. The relationships of the triplets can be verbs (e.g., sits on, wears) or prepositions (e.g., on, with) describing relations between objects in an image. Similar to current research developments [2][9][10][11], the datasets with the 100 and 50 most common relationships are used.

2

The generation of the scene graph generation consisted of three phases: 1) Object collection using the ConceptNet API; 2) spatial relation prediction; 3) scene graph generation using RDFLib [12].

For step 2, three different models are compared: a statistical model based on most frequent relations, a ConceptNet-trained KnowBERT model, a Wikipedia-WordNet KnowBERT model. For the last two models only the most certain predictions were kept in the dataset.

Five scene graphs were generated to evaluate each of the three models: a garden, a bathroom, a living room, a bedroom, and a kitchen.

Results. While the method is successful at creating common-sense graphs, some wrong relations were predicted using the KnowBERT models.
For the spatial relation prediction, the Wikipedia-WordNet model outperformed the ConceptNet model slightly in the 100-relation model (F1 = 0.55, F1 = 0.51± 0.01) but not for the 50-relation model (F1 = 0.54± 0.03, F1 = 0.54± 0.01). This could be due to the fact that the Wikipedia + WordNet model is trained on two knowledge bases. Wikipedia contains a lot of textual information on an entity, while WordNet synsets give information on what entities are related. The statistical model proved to be slightly superior over both KnowBERT models, with an accuracy of 0.59 ± 0.01 for the 100-relation model and an accuracy of 0.62 ± 0.04 for the 50-relation model. However, for unseen relations, all KnowBERT models perform far better than the statistical model. The model for 100 relations has an accuracy of 0.53 and 0.48 for the Wikipedia-WordNet and the ConceptNet model respectively against an accuracy of 0.29 for the statistical model. The model for 50 relations has an accuracy of 0.53 and 0.56 for the Wikipedia-WordNet and the ConceptNet model respectively against an accuracy of 0.36 for the statistical model. To conclude, BERT can be combined with several knowledge bases to create common sense graphs.

References

1. Zareian, Alireza, Svebor Karaman, and Shih-Fu Chang. "Bridging knowledge graphs to generate scene graphs." In *European Conference on Computer Vision*, pp. 606-623. Springer, Cham (2020).
2. Chen, T., Yu, W., Chen, R., & Lin, L. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6163-6171 (2019).
3. Gu, J., Zhao, H., Lin, Z., Li, S., Cai, J., & Ling, M. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1969-1978) (2019).
4. Miller, G. A.. WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39-41. (1995).
5. Liu, H., & Singh, P. ConceptNet—a practical commonsense reasoning tool-kit. *BT technology journal*, *22*(4), 211-226. (2004).
6. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* (2019).
7. Peters, M. E., Neumann, M., Logan, R. L., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. Knowledge enhanced contextual word representations. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference* (2020).
8. Krishna, Ranjay, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen et al. "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations." *International Journal of Computer Vision* 123, no. 1 (2017).
9. Yu, R., Li, A., Morariu, V. I., & Davis, L. S. Visual Relationship Detection with Internal and External Linguistic Knowledge Distillation. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-October*, 1068–1076 (2017).
10. Zhang, H., Kyaw, Z., Chang, S.-F., & Chua, T.-S. *Visual Translation Embedding Network for Visual Relation Detection* (2017).
11. Zellers, R., Yatskar, M., Thomson, S., & Choi, Y. Neural Motifs: Scene Graph Parsing with Global Context. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 5831–5840 (2018).
12. rdflib 5.0.0 — rdflib 5.0.0 documentation. (n.d.). Retrieved June 28, 2021, from https://rdflib.readthedocs.io/en/stable/

Localised Reputation in the Prisoner's Dilemma

Martin Toman and Neil Yorke-Smith[0000−0002−1814−3515]
m.toman@student.tudelft.nl,n.yorke-smith@tudelft.nl

Delft University of Technology, The Netherlands

Abstract. Under what conditions can cooperation emerge and be sustained? Previous studies abstract cooperation and defection using the spatial Prisoner's Dilemma (PD) game. We study a local reputation mechanism in which agents can remember defectors, abstain from interacting with them, and warn nearby agents. Simulations find that local reputation is effective in sustaining cooperation and punishing defection. Further, we find that the size of agent memory and amount of gossip are not significant factors, provided that the locality range of gossip is greater than the agent movement speed.

1 Motivation and Experimental Design

Reputation systems strongly boost cooperation in spatial exchange games such as spatial PD [2, 6]. Similarly, allowing game participants to pass information, either directly [4] or indirectly [1], increases the rate of cooperation.

We aim to explore the limits of local reputation—built up via gossip—in promoting and sustaining cooperation. Agent's behaviour is defined by the finite state diagram shown in Figure 1. We expand over prior work [5] by giving agents a (limited size) memory to keep track of defectors and to allow them to share this information by gossiping with other agents in a certain range.

2 Results and Discussion

Agents are one of two types: cooperator or defector. We allow agents to remember the five most recent defectors and to ask nearby agents in a Moore neighbourhood of radius 1, 2 and 3 if they remember an agent defecting in a certain number of

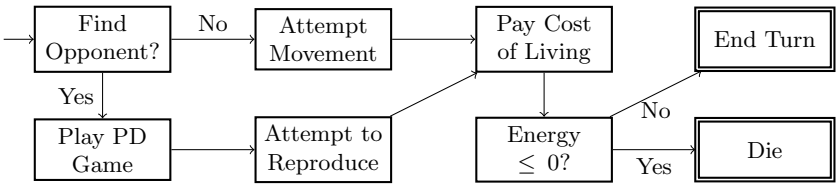


Fig. 1. Agent behaviour diagram: showing the decision flow of an agent's single turn

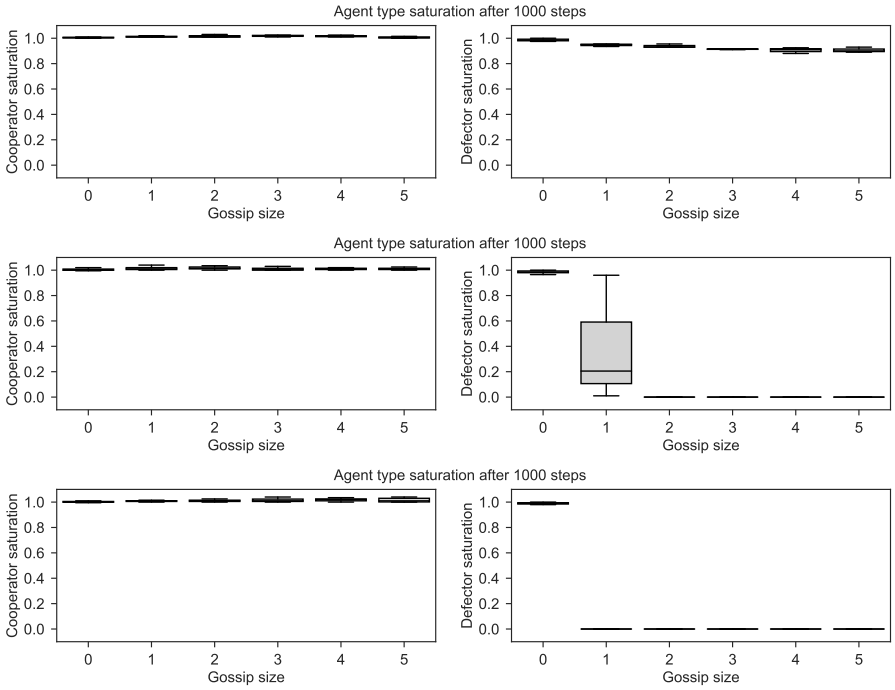


Fig. 2. Cooperator agent saturation for various gossip sizes after 1000 steps. Left column: no gossip, right column: with gossip mechanism. Top row to bottom: gossip radii 1, 2 and 3, respectively. Std. dev. of 30 simulation runs, outliers removed

past encounters—varying between 0 and 5. We run the simulation for 1000 steps and plot the saturation percentage of cooperator agents in Figure 2.

The introduction of gossip is a strong deterrent of defection and quickly leads to cooperator-only populations, as seen in the right column. We find that the size of the memory and the size of the gossip are not significant factors, only speeding up the convergence slightly.

Our simulation results also find that the most important factor in predicting cooperator success is the range at which gossip can be exchanged; the amount of information included in the gossip has negligible effect. If the gossip can move faster than agents, cooperators will flourish. Otherwise, defectors can reach full population saturation.

We studied a local reputation mechanism in spatial PD. Several directions can build on our results. Notably, we assumed all information is transferred with 100% fidelity. However, not all strategies that perform well in noiseless environments can do so under the presence of noise [3]. If the agent behaviour is unpredictable enough, the gossip mechanism could deter more cooperator–cooperator interactions: the pros and cons in noisy environments deserve investigation.

References

[1] Camera, G., Casari, M.: Cooperation among strangers under the shadow of the future. American Economic Review **99**, 979–1005 (2009). https://doi.org/10.1257/aer.99.3.979

[2] Dong, Y., Sun, S., Xia, C., Perc, M.: Second-order reputation promotes cooperation in the spatial prisoner's dilemma game. IEEE Access **7**, 82532–82540 (2019). https://doi.org/10.1109/ACCESS.2019.2922200

[3] Gevers, L., Yorke-Smith, N.: Cooperation in harsh environments: The effects of noise in iterated prisoner's dilemma. Proceedings of BNAIC/BeneLearn 2020 pp. 414–415 (2020)

[4] Kagel, J.H.: Cooperation through communication: Teams and individuals in finitely repeated prisoners' dilemma games. Journal of Economic Behavior & Organization **146**, 55–64 (2018). https://doi.org/https://doi.org/10.1016/j.jebo.2017.12.009

[5] Smaldino, P.E., Schank, J.C., McElreath, R.: Increased costs of cooperation help cooperators in the long run. The American Naturalist **181**(4), 451–463 (2013). https://doi.org/10.1086/669615

[6] Stahl, D.O.: An experimental test of the efficacy of a simple reputation mechanism to solve social dilemmas. Journal of Economic Behavior & Organization **94**, 116–124 (2013). https://doi.org/https://doi.org/10.1016/j.jebo.2013.08.014

Remote NO_2 emissions assessment during COVID-19 lockdowns

Abigail Vella[1[0000−0003−0391−5911]], Frankie Inguanez[1[0000−0001−8396−4443]], and Daren Scerri[1[0000−0002−2516−8972]]

Institute of Information & Communication Technology
Malta College of Arts Science & Technology
Paola PLA9032, Malta
abigail.vella.b42203@mcast.edu.mt, frankie.inguanez@mcast.edu.mt, daren.scerri@mcast.edu.mt

Abstract. This study researches the changes in NO_2 concentrations over the Maltese islands by comparing the readings gathered from the Sentinel-5P satellite, pre-COVID-19 and during COVID-19 for the months of March, April and May 2019 and 2020. It was found that during the lockdown period, NO_2 levels dropped by 12.79% in March, 14.39% in April but increased by 7.31% in May as lockdown restrictions started to be eased. When comparing the results gathered from Sentinel-5P with the World Air Quality Index (WAQI) values obtained by the local Environment Resource Authority (ERA), a correlation of 98% for the monthly delta values were found. Comparing the remote data for Malta with other close major European cities also showed a similar correlation.

Keywords: Remote Sensing · COVID-19 · Sentinel-5P · NO_2

1 Introduction

The monitoring of air quality is a national priority as well a regional (EU) priority, to monitor the improvement of the quality of life for human beings. There are several approaches to monitor air quality, such as, local land sensors and remote sensing. Land based assessment has been exclusively used in the Maltese islands since 2016 with a total of four sensors (Attard, Għarb in sister island Gozo, Msida and Żejtun) spread across the islands to cover an area of 316Km^2 [3]. This research focuses on reviewing the viability of using Sentinel-5P data to determine a better product regarding quality, that can be offered.

2 Results

A total of 235 products were downloaded from Sentinel-5P over a period of 184 days taking up 88.1GB of storage. The land sensors actually provide a continuous hourly based reading, which the ERA make available on their portal and aggregated on the WAQI portal. It is important to note that the NO_2 units

from the WAQI dataset are calculated in $\mu g/m^3$, while the Sentinel-5P level 2 are calculated in mol/m^2. For further analysis on the Sentinel-5P values, the NO_2 averages were compared for March, April, and May 2019 and 2020 for the three regions (Gozo, North and South). It was observed that the NO_2 values

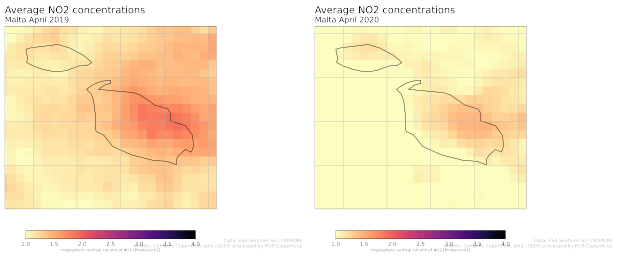


Fig. 1. Sentinel-5P Level 3 Images April 2019 & April 2020

were lowest in the region of Gozo, which is more rural then Malta, through all three months for both 2019 and 2020. When looking at the NO_2 values for the North and South regions, overall the North region had higher NO_2 values for both 2019 and 2020. WAQI values show that, NO_2 reduced by 49.51% between 2019 and 2020, and according to the Sentinel-5P values, NO_2 reduced by 6.24% between 2019 and 2020. Due to different units of measure adopted from different datasets, we cannot directly compare these datasets and [2] do not recommend converting to compare. The Harp library was used to convert the daily satellite Level 2 images to monthly aggregated level 3 composite visualisation. Passive assessment maps from ERA were visually compared with the Sentinel-5P level 3 composite, shown in Figure 1, and we could see a direct correlation. The delta in each dataset was then statistically calculated and correlation across the 2 datasets which was found to be 98% well over the 75% standard value in this sector of research, which indicates that both averages from both sensors will calculate the same percentage drop/increase in NO_2.

In $[5, 6, 1, 4]$, the majority of NO_2 reductions for major cities in Europe were between the months of March, April and May, ranging between 22% - 54% from values gathered from Sentinel-5P and 17% - 50% from values gathered from *in situ* observations. Although the difference in Malta is small it is expected since Malta is less densely populated than Milan, having the population densities of 1,383 people/km^2 and 7,551 people/km^2 respectively as of 2018.

3 Conclusion

The remote sensing assessment has proven to be a reliable and consistent source of data, currently unexplored and underutilised at a local level. The findings of this research have been presented to the *Environment & Resource Authority* (ERA), who consult the local government, for their consideration.

References

1. Barré, J., Petetin, H., Colette, A., Guevara, M., Peuch, V.H., Rouil, L., Engelen, R., Inness, A., Flemming, J., Pérez García-Pando, C., et al.: Estimating lockdown induced european no2 changes. Atmos. Chem. Phys. Discuss.[preprint], https://doi.org/10.5194/acp-2020-995, in review (2020)
2. Borsdorff, T., Hu, H., Hasekamp, O., Sussmann, R., Rettinger, M., Hase, F., Gross, J., Schneider, M., Garcia, O., Stremme, W., et al.: Mapping carbon monoxide pollution from space down to city scales with daily global coverage. Atmospheric Measurement Techniques **11**(10), 5507–5518 (2018)
3. Environment and Resources Authority: A preliminary assessment related to the impact of covid-19 measures on air quality in malta (2020)
4. Mesas-Carrascosa, F.J., Pérez Porras, F., Triviño-Tarradas, P., García-Ferrer, A., Meroño-Larriva, J.E.: Effect of lockdown measures on atmospheric nitrogen dioxide during sars-cov-2 in spain. Remote Sensing **12**(14), 2210 (2020)
5. Muhammad, S., Long, X., Salman, M.: Covid-19 pandemic and environmental pollution: A blessing in disguise? Science of the total environment **728**, 138820 (2020)
6. Vîrghileanu, M., Săvulescu, I., Mihai, B.A., Nistor, C., Dobre, R.: Nitrogen dioxide (no2) pollution monitoring with sentinel-5p satellite imagery over europe during the coronavirus pandemic outbreak. Remote Sensing **12**(21), 3575 (2020)

Automated Negotiation Under User Preference Uncertainty

Adel Magra, Peter Spreij[1], Tim Baarslag[2], and Michael Kaisers[2]

[1] Korteweg-de Vries Institute for Mathematics, University of Amsterdam
[2] Centrum Wiskunde & Informatica, Amsterdam

1 Introduction

We are concerned with automated agents representing humans in negotiations. To negotiate effectively and obtain a favorable outcome, the agent must know the preferences of the human user it is representing. These preferences are often represented by a utility function. When the agent does not know these preferences, we say that it is negotiating under uncertainty (Fig 1). To gather information about these preferences, the agent can interact with the user by asking questions or queries. The whole point of automating a negotiation is to make it more convenient for the user, we therefore do not want the agent to ask too many queries. Optimal queries were previously considered as ones with high expected value of information, which is the prospected gain in utility that a query can add to the final outcome of the negotiation (i.e. the one agreed upon) [1]. We bring forward another perspective: We consider queries as optimal based on their inherent ability to reduce uncertainty on the user's preferences.

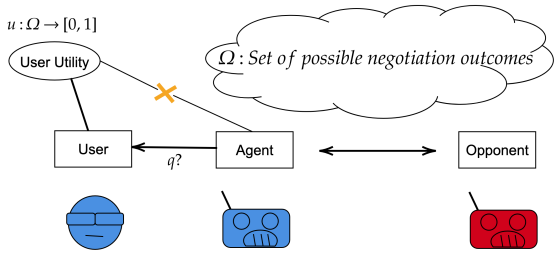


Fig. 1. Negotiating Under User Preference Uncertainty

2 A Framework for Optimal Reduction of Uncertainty

We build a general framework to formally deal with the problem of reducing uncertainty (Fig 2). We assume that the true user utility is parametrizable by $\theta^* \in \Theta$ and that queries can be answered by the user according to an answer function a. We introduce the notion of information potential of a query, which is the minimal amount of information that the agent can extract on the user's utility by asking it. It quantifies the worst possible reduction of uncertainty that is obtained when asking a query. Formally, we denote it by $I(q)$ for a query q and define it as such:

$$\begin{aligned} &I(q) := \min_{r \in \mathcal{A}} -\log \Pi(\Theta_{(q,r)}) \\ &\textit{where } \forall r \in \mathcal{A},\ \Theta_{(q,r)} := \{\theta \in \Theta : a(\theta^*, q) = r\} \end{aligned} \tag{1}$$

Based on its current belief on the user preferences, the agent's objective therefore becomes to ask a query that maximizes the information potential. After observing the answer to a query $a(\theta^*, q)$, the agent's belief is narrowed down to the set $\Theta_{(q,a(\theta^*,q))}$, which we call the posterior set. We thus provide an objective approach of reducing uncertainty through a sequential optimization problem: the agent must find a sequence of queries maximizing the information potential.

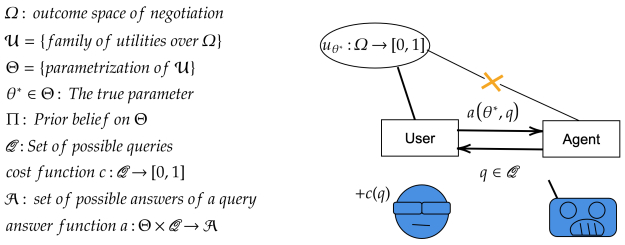


Fig. 2. Our Framework to Deal with User Preference Uncertainty

3 Application to Multi-Issue Negotiation

A multi-issue negotiation domain is divided into different issues whose combinations form possible deals for the negotiation: $\Omega = I_1 \times \cdots \times I_n$. A popular way of representing a user's preferences on a multi-issue domain is through a linear additive utility function, which gives relative importance weights to issues and valuations for the possible values of issues. The unknown user utility is: $u_{\theta^*}(\omega) = \sum_{i=1}^{n} \theta_i^* \cdot \text{val}(\omega_i)$, with θ^* being in the standard $n-1$ simplex Δ^{n-1}, where $\Delta^{n-1} = \{\theta \in \mathbb{R}^+, \sum_{i=1}^{n} \theta_i^* = 1\}$. The queries we consider are pairwise outcome comparisons: asking the user to compare two given outcomes in Ω.

We use our framework to derive an optimal querying algorithm to reduce uncertainty on the issue weights vector θ^*. We assume a uniform prior on Δ^{n-1}. Because u_{θ^*} is linear additive, pairwise outcome comparisons correspond to hyperplanes. Our Optimal Query Sequence (OQS-n) algorithm (Algorithm 1) exploits that by successively finding a query that bisects the current posterior set. Under some assumptions on the valuation functions, we show that OQS-n generates query sequences of absolutely maximal information potential of arbitrary length T (Theorem 1).

Algorithm 1: OQS-n

Input: $\Omega, \text{val}_1, \cdots, \text{val}_n, T$

$P \leftarrow V(\Delta^{n-1})$ // Store the n vertices of Δ^{n-1}

for $t \in \{1, \cdots, T\}$ **do**

  $(p, q) \leftarrow_{(p_i, p_j) \in P^2} d(p_i, p_j)$ // Find longest edge of P

  $m \leftarrow \frac{1}{2}(p + q)$ // mid point of longest edge

  $\ell \leftarrow \mathcal{HP}(P \backslash \{p, q\}, m)$ // ℓ is the hyperplane bisecting P

  $q \leftarrow Query(\ell)$ // Find a query corresponding to ℓ

  $a \leftarrow Ask(q)$

  $P \leftarrow Update(q, a)$

Theorem 1. *For any given $\theta^* \in \Delta^{n-1}$, and any length $T \in \mathbb{N}$, OQS-n (Algorithm 1) produces a query sequence of length T of maximal information potential.*

References

1. Baarslag, Tim and Kaisers, Michael: The value of information in automated negotiation: a decision model for eliciting user preferences. In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, May 2017, pp 391-400.

Thesis Abstract: Interactive Subgroup Discovery for the Conversational Data Governance Platform "Talking to your Data"*

Astrid Sierens[1], Isel Grau[1][0000−0002−8035−2887], Luis Daniel Hernandez[1], Simeon Michel[2], Vicky Froyen[3][0000−0002−5649−5888], Catherine Middag[2][0000−0001−5732−0281], and Ann Nowe[1][0000−0001−6346−4564]

[1] Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium
[2] Gezondheidszorg, Design & Technologie, Erasmushogeschool Brussel, Belgium
[3] Collibra NV, Belgium

Abstract. In this master thesis, a new interactive subgroup discovery algorithm is proposed. This method has two main contributions. First, the algorithm allows the expert to intervene during the search process by assessing each subgroup with a degree of appreciation which influences the search process. The second contribution is a diversity parameter that allows the user to avoid that the new subgroups share more than a chosen percentage of instances with already found subgroups. Experiments show that when diversity control is performed, the resulting subgroups have less overlap than the baseline version of the algorithm. Additionally, when using the proposed interactive version of the algorithm, a higher user appreciation of the subgroups is observed. This interactive subgroup discovery algorithm was implemented in the backend of a conversational agent for supporting business analysts in data mining tasks.

Keywords: subgroup discovery · interactive machine learning · explainable artificial intelligence

1 Introduction

Subgroup discovery [10] is a data mining technique that lies between predictive and descriptive analysis. These algorithms search for subsets of data points that are characterized by a value of interest with respect to the target feature and also share similar properties within the subsets. For example, a company might be interested in the characteristics of the subgroup of customers for which a given advertisement campaign was successful. Subgroup discovery algorithms mainly differ in their search strategy for generating candidate subgroups and the quality measures they use for ranking the subgroups.

The description of these subgroups or local patterns can be done in the form of rules, which makes the resulting model intrinsically interpretable, similarly

* Supported by the Innoviris TeamUp project "Driving collective data governance through smart engagement platforms".

to other rule-based models [4, 9]. A rule describing a subgroup has the form *Cond* $\longrightarrow$ *Target* where *Cond* is the condition, often consisting of a conjunction of feature-value pairs, and *Target* is the value for the variable of interest. An example of such a rule or subgroup is the following: (*Salary* $> 80K$ *AND Education* = *University*) $\longrightarrow$ *Loan Approved* = *Yes*. This subgroup describes that the population group with a high income and education level is more likely to receive an approved loan compared to the total population.

Although several algorithms for subgroup discovery are proposed in the literature [6, 5], these methods often produce overlapping and therefore redundant subgroups [1]. Furthermore, they often produce general and obvious rules which are already known by the user, therefore they are regarded as uninteresting [3]. To overcome these drawbacks, we propose two modifications that allow for more interactivity and diversity in the resulting subgroups. The first contribution is that our algorithm allows the user to intervene during the search process by assessing each subgroup with a degree of appreciation. This weight is used to adjust the quality measure of the subgroup and thus influences the search process. The second contribution is a diversity parameter that allows the user to avoid that the new subgroups share more than a chosen percentage of instances with already found subgroups.

2 Experiments and Results

We implemented the proposed interactive algorithm in the context of the data mining backend for the conversational platform "Talking to your Data" [8, 7]. The goal of this conversational agent is to bring the data mining process closer to business analysts by translating their needs expressed as natural language to data mining tasks, execute them and translating back the generated rules to natural language. This facilitates the analysis of data and generating explanations in a conversational way that is appropriate for the audience in question [2].

We performed several experiments involving three datasets and a profile of a hypothetical user that generally does not like common knowledge subgroups. These experiments show that when diversity control is performed, the resulting subgroups have less overlap than the baseline version of the algorithm. Additionally, when using the proposed interactive version of the algorithm, a higher user appreciation of the subgroups is observed. However, from the experiments where both interactivity and diversity are applied, there is not a big gain from applying diversity on top of interactivity. This is especially the case if the user profile values more the rare and interesting subgroups than the large and well-known subgroups.

Future work on this topic would include an extension of the experimentation with other profiles of user behavior (e.g. more interested in general rules) or a real-world scenario to conduct a user test with the participation of experts. Extending the implementation with extra support to guide the user in their choice and include extra measures to evaluate the entire subgroup set as a whole would also be useful.

References

1. Atzmueller, M.: Subgroup discovery. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **5**(1), 35–49 (2015). https://doi.org/10.1002/widm.1144
2. Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012
3. Dzyuba, V., van Leeuwen, M.: Interactive discovery of interesting subgroup sets. In: International Symposium on Intelligent Data Analysis. pp. 150–161. Springer (2013)
4. Grau, I., Sengupta, D., Garcia Lorenzo, M.M., Nowé, A.: An Interpretable Semi-supervised Classifier using Rough Sets for Amended Self-labeling. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE (2020)
5. Helal, S.: Subgroup Discovery Algorithms: A Survey and Empirical Evaluation. Journal of Computer Science and Technology **31**(3), 561–576 (2016). https://doi.org/10.1007/s11390-016-1647-1
6. Herrera, F., Carmona, C.J., González, P., del Jesus, M.J.: An overview on subgroup discovery: Foundations and applications. Knowledge and Information Systems **29**(3), 495–525 (2011). https://doi.org/10.1007/s10115-010-0356-2
7. Loeckx, J., Grau, I., Sergeyssels, N., Michel, S., Froyen, V., Middag, C., Nowe, A.: Driving Collective Data Governance through Smart Engagements Platforms (Collibra) (2018), https://ai.vub.ac.be/portfolio/driving-collective-data-governance-through-smart-engagement-platforms/
8. Loeckx, J., Keizer, S., Grau, I.: VUB AI Learning and Intelligent SlackBots (2018), https://university.collibra.com/learn/course/external/view/elearning/60/vub-ai-learning-and-intelligent-slackbots
9. Molnar, C.: Interpretable Machine Learning. Leanpub (2019)
10. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **1263**, 78–87 (1997). https://doi.org/10.1007/3-540-63223-9$_1$08

Pain recognition from thermal videos using deep neural networks

Aleksandra Olczyk, Itir Onal Ertugrul

Department of CSAI, Tilburg University, Tilburg, the Netherlands
a.m.olczyk@tilburguniversity.edu
I.onal@tilburguniversity.edu

Abstract. This thesis investigated the performance of different neural network architectures on a facial expression-based pain recognition task. In addition, the efficacy of transfer learning between the RGB and thermal domain was assessed. To achieve that, the BP4D+ dataset, which contains thermal videos of 140 subjects experiencing ten different emotions, was used. It was found that the best performing model consisted of CNN that is pre-trained on RGB images and finetuned on thermal images combined with LSTM, however the performance gap between the other models was not extreme.

Keywords: Pain recognition · Thermal imaging · Deep learning

1 Introduction

Pain is a prevalent medical and societal problem. Historically, self-report or observer implemented scales were the gold standard of pain assessment. However, patients and doctors tend to evaluate pain levels differently [1] and self-report methods cannot be reliably used by individuals with cognitive impairments, unconscious patients, or children [2]. This warrants a development of more objective and automatic pain assessment methods. Facial expression has a lot of potential in the context of pain recognition, however most of the work in this field focused on facial expression of pain captured in visible spectrum (e.g., red-green-blue referred to as RGB) domain and little attention has been dedicated to the use of thermal imaging. It has certain advantages over both traditional RGB images and physiological measurements such as being insensitive to illumination [3], correlating with other physiological signals [4, 5] and having the potential to minimize the privacy concerns.

Automatic detection and intensity estimation of pain have been mostly explored through the McMaster-UNBC Shoulder Pain Archive Database [6]. Conventional models such as Support Vector Machines [7, 8], AdaBoost [9], Hidden Markov Models [10], as well as deep learning methods [11] were explored. Deep neural networks proved to be particularly effective. Using 2D CNN + LSTM architecture, Rodriguez et al. [11] achieved 93.3% AUC score, outperforming the previous top AUC score of 84.7% [12]. In contrast, the literature on the use of thermal data for emotion recognition, and pain recognition in particular, is very scarce. Lack of research in this area can be partially explained by the absence of well-annotated datasets in other visual modalities. Zhang et al. [13] developed a multi-modal spontaneous emotion corpus, BP4D+, which contains facial expression data in RGB, 3D and thermal modalities, alongside physiological data. The authors validated the thermal data subset on a facial emotion recognition task, achieving 91% accuracy using SVM, demonstrating the utility of thermal data in the context of emotion recognition task. The present study provided a three-fold contribution to the scientific community: establishing a baseline performance on pain recognition on the thermal subset of BP4D+ dataset, comparing the results of

CNNs trained exclusively on thermal data and models pre-trained on RGB images, and lastly, comparing the performance of LSTMs and GRUs on pain recognition using thermal imaging.

2 Methods

The architectures explored in this study combined two networks: 2D CNN that was used to obtain spatial representation in each frame, and RNN, either LSTM or GRU, used to model temporal information in a sequence of frames. The CNN selected was VGG-16 [14], as it was shown to give promising results on other emotion recognition tasks [11, 15]. The CNN was (i) pre-trained on ImageNet and fine-tuned with the thermal subset of BP4D+ dataset or (ii) trained from scratch exclusively on thermal data. Both versions were trained using the same settings: stochastic gradient descent optimizer with learning rate of 0.001, momentum of 0.9, mini-batch sizes of 32 and cross-entropy loss. Once the CNNs were trained, feature vectors of length 4096 were extracted from the *fc6* layer as in [11]. From each video only 16 frames were selected and passed through the CNN network. These were then concatenated to from an array of shape 16x4096.

A grid search was performed to find the best set of hyperparameters of LSTM and GRU. Adam optimiser was used [16], with different training rates explored. To ensure better generalisability of the model and to potentially reduce overfitting, two regularization strategies were used: applying L-2 regularization and introducing a drop-out layer with varying drop-out rate. Twelve different combinations of hyperparameters were compared and all the models were trained for 25 epochs. To choose the best performing model, three different evaluation metrics were used: weighted accuracy, AUC and F-1 score of both the minority and majority class.

3 Results

CNN fine-tuned on thermal images combined with LSTM outperformed all the other models, achieving weighted accuracy of 84.37%, AUC score of 0.84, F-1 score of 0.55 and 0.92 for the “pain” and “no pain” classes, respectively. Interestingly, CNN trained from scratch performed best with GRU, however the difference between GRU and LSTM models was not overwhelming. Finally, models based on fine-tuned CNN made more false positives compared with false negatives, whereas the opposite was the case for models based on CNN trained from scratch.

Model	Weighted Accuracy	AUC	F-1 pain	F-1 No pain
Fine-tuned CNN + LSTM 256 hidden size, 1 layer, lr = 0.0001, dr = 0.4	84.37	0.84	0.55	0.92
CNN trained from scratch + GRU 100 hidden size, 1 layer, lr = 0.0001, wd = 0.0001, dr = 0.4	78.05	0.74	0.55	0.96

Table 1. Results on the test set of two best performing models.

References

1. Hammal, Z. and J.F. Cohn, *Automatic detection of pain intensity.* Proceedings of the ... ACM International Conference on Multimodal Interaction. ICMI (Conference), 2012. **2012**: p. 47-52.
2. Herr, K., et al., *Pain assessment in the patient unable to self-report: position statement with clinical practice recommendations.* Pain Management Nursing, 2011. **12**(4): p. 230-250.
3. Nguyen, T., K. Tran, and H. Nguyen. *Towards Thermal Region of Interest for Human Emotion Estimation*. in *Conference: 2018 10th International Conference on Knowledge and Systems Engineering (KSE*. 2018.
4. Pavlidis, I., et al., *Interacting with human physiology.* Computer Vision and Image Understanding, 2007. **108**(1): p. 150-170.
5. Sonkusare, S., et al., *Detecting changes in facial temperature induced by a sudden auditory stimulus based on deep learning-assisted face tracking.* Scientific Reports, 2019. **9**(1): p. 4729.
6. Lucey, P., et al. *Painful data: The UNBC-McMaster shoulder pain expression archive database*. in *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. 2011.
7. Chen, J., Z. Chi, and H. Fu, *A new framework with multiple tasks for detecting and locating pain events in video.* Computer Vision and Image Understanding, 2017. **155**: p. 113-123.
8. Kharghanian, R., A. Peiravi, and F. Moradi, *Pain detection from facial images using unsupervised feature learning approach*. Vol. 2016. 2016. 419-422.
9. Lo Presti, L. and M. La Cascia, *Boosting Hankel matrices for face emotion recognition and pain detection.* Computer Vision and Image Understanding, 2017. **156**: p. 19-33.
10. Meng, H. and N. Bianchi-Berthouze, *Affective State Level Recognition in Naturalistic Facial and Vocal Expressions.* IEEE transactions on cybernetics, 2013. **44**.
11. Rodriguez, P., et al., *Deep Pain: Exploiting Long Short-Term Memory Networks for Facial Expression Classification.* IEEE transactions on cybernetics, 2017.
12. Lucey, P., et al., *Automatically Detecting Pain in Video Through Facial Action Units.* Trans. Sys. Man Cyber. Part B, 2011. **41**(3): p. 664–674.
13. Zhang, Z., et al. *Multimodal Spontaneous Emotion Corpus for Human Behavior Analysis*. in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
14. Simonyan, K. and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition.* arXiv 1409.1556, 2014.
15. Li, Y., et al., *Occlusion Aware Facial Expression Recognition Using CNN With Attention Mechanism.* IEEE Transactions on Image Processing, 2019. **28**(5): p. 2439-2450.
16. Kingma, D. and J. Ba, *Adam: A Method for Stochastic Optimization.* International Conference on Learning Representations, 2014.

Safe Fleet-Wide Policy Iteration

Domien Hennion, Timothy Verstraeten, and Ann Nowé

Vrije Universiteit Brussel, Brussels, Belgium
{domien.hennion, timothy.verstraeten, ann.nowe}@vub.be

In many settings, multiple devices with similar design specifications are instantiated to execute the same control task, which is called a *fleet*. An example of such a setting is a wind farm, in which a group of wind turbines aim to optimize power production. With the Industry 4.0, cyberphysical machines are equipped with modern wireless sensors, while their data is being transmitted to a cloud-based architecture. This allows a fleet of machines to be monitored and controlled as a single system. In order to improve the efficiency and optimality of the fleet controller, it is important to exploit the similarities between the machines and establish a framework through which data can be shared. Fleet reinforcement learning tackles this setting, and uses data exchange in order to improve the control task of multiple reinforcement learning agents operating in similar environments . However, the safety issue of applying reinforcement learning in fleet settings has not been addressed yet. Specifically, allowing multiple machines to randomly explore the environment while learning may violate physical constraints and potentially damages the machines. Therefore, we propose a novel fleet reinforcement learning algorithm that uses a safe exploration mechanism. Specifically, We ensured safety for the exploration phase of this algorithm by implementing a Control Barrier Function (CBF) (Cheng, Orosz, Murray, & Burdick, 2019). A CBF blocks unsafe actions by modeling the unknown system dynamics of the agent's environment through a Gaussian process.

To examine the safety of our implementation, we experimentally analyzed the CBF on a fleet variant of the mountain car benchmark (Moore, 1990), containing 3 mountain cars with varying mass. The target mountain car needs to estimate a sufficiently accurate transition model by transferring knowledge based on learned correlations with the source members. When an optimal policy is found, it is executed greedily and safely with the implemented CBF.

Figure 1 visualizes the agent's behaviour. η represents how strongly the CBF restricts the agent into a set of safe states, with $\eta = 1$ being the most unsafe version of the CBF and $\eta = 0$ being the safest version. A reoccurring strategy of the agent is removing the unsafe exploration of different actions. We observe that $\eta = 1$ leads to a safer approach to find the optimal strategy, whereas $\eta = 0$ leads to an optimal solution in a slower and preventive manner. This demonstrates the effects of the CBF.

We also demonstrated our method on the FLORIS state-of-the-art wind farm simulator . Our results show that in this case the agent also adopts a safer control policy with our proposed CBF-based fleet control algorithm.

Still, riskier actions may often lead to higher rewards. Therefore, it is expected that incorporating the CBF introduces a decrease in performance. Figure 2 shows

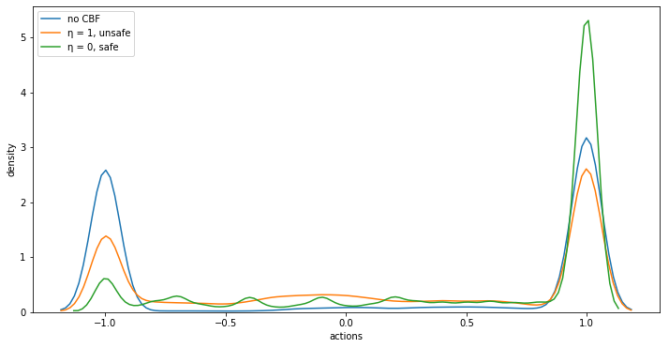


Fig. 1. Density mountain car action usage. Action -1 driving backwards and 1 driving forward

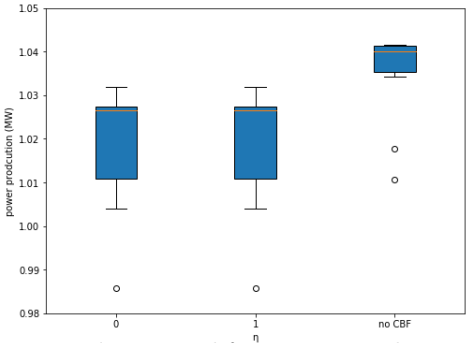


Fig. 2. Wind farm power production

how this loss in performance translates into the power production of a wind farm. Specifically, the median power production decreased from 1.04 MW to 1.026 MW.

The trade-off between performance and safety is a necessary decision to be made by the fleet's operators. Our algorithm uses the η parameter through which the operators can adapt this trade-off in a transparant manner.

Our experimental results showed the potential benefits of our safe reinforcement learning algorithm in real-world fleet applications. We demonstrated that our safe fleet-wide policy iteration method can ensure safety while still minimizing the performance gap with the unsafe version.

References

Cheng, R., Orosz, G., Murray, R. M., & Burdick, J. W. (2019). End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 3387–3395).

Moore, A. W. (1990). Efficient memory-based learning for robot control.

NREL. (2019). *FLORIS. Version 1.0.0.* GitHub. Retrieved from https://github.com/wisdem/floris

Verstraeten, T., Libin, P. J., & Nowé, A. (2020). Fleet control using coregionalized gaussian process policy iteration. *Proc. of the 24th European Conference on Artificial Intelligence (ECAI) (in press)*.

Master's thesis abstract: "Bias quantification measures based on fuzzy rough sets"

Lisa Koutsoviti Koumeri[1,2]
Supervisor: Dr. Gonzalo Nápoles[2]

[1] Hasselt University, Martelarenlaan 42, 3500 Hasselt, Belgium
[2] Tilburg University, Warandelaan 2, 5037 AB Tilburg, Netherlands

Motivation. Artificial Intelligence (AI) systems are widely employed to solve pattern classification problems. These often include classifying which people can get a loan, receive medical treatment, or commit a crime. These life-changing decisions should be fair, i.e. not be based on protected features like race or gender. However, research revealed that this is not always the case due to biased labels or imbalanced data. Aiming to tackle this issue, numerous bias measures have been proposed, but despite these efforts, there is still great need to introduce new measures [1] for the following reasons. Existing approaches depend on different and often conflicting notions of fairness [4] or might consider part of the information available in the dataset (only sensitive and target features) [2]. Moreover, they often depend on black-box Machine Learning (ML) models whose outputs are sensitive to data preprocessing or training-test splits, and are not intuitively explainable. Finally, users need to make assumptions regarding the discriminated feature-category. We attempt to offer a remedy to these challenges.

This thesis proposes five measures based on the fuzzy-rough set (FRS) theory to quantify bias related to sensitive features of pattern classification datasets. This mathematical theory allows analyzing inconsistency in decision making systems [7], can define similarity thresholds when handling continuous features [6] while offering an explainable semantic background.

Methods. The measures are computed in a two step process. As a first step, we build information granules describing each decision class following the FRS formalization as introduced by [8]. Three information granules are computed per decision class: a positive, negative and boundary fuzzy-rough region. The membership value of an instance to a certain positive region indicates the extent to which the instance belongs to a decision class, does not belong to that class or the extent to which the instance belongs to the boundary region. This fuzzy granulation process is repeated twice: first, the three fuzzy-rough regions are calculated using all features in the data and, next, they are calculated again *excluding one of the protected features*. The intuition is that removing features from the decision making process should not cause large changes in the fuzzy-rough regions. The extent to which this happens is a proxy for bias.

As a second step, the five measures are calculated. These measures quantify the change in the membership values characterizing fuzzy-rough regions after the suppression of a protected feature. The first two measures quantify the change locally (between decision classes and information granules) and the rest glob-

ally (between information granules). Note that these values are not absolute but should be interpreted relatively to the respective values reported when we suppress a different protected feature. If the measures report relatively larger values regarding a certain protected feature, then that means that the exclusion of this same feature has a greater impact on the classification process, which can be understood as evidence for explicit bias.

Numerical simulations. The proposed measures are tested on *German Credit* and *Compas* datasets [5]. Protected features are *age* and *gender* for the former and *race* and *gender* for the latter. Decision classes are *creditworthy* or the opposite and *likely* or *unlikely to re-offend* respectively. The outputs of the proposed fuzzy-rough measures are compared to four popular bias measures [3] that fall under the category of group fairness and are computed using the AIF360 open source toolkit [5]. Results showed that almost all proposed measures differ from the literature measures both in direction and magnitude (a sample of the results is shown in Table 1). Such a disagreement raises concerns regarding the consistency of measures for bias quantification.

Table 1: Results of baseline and global fuzzy-rough measures tested on *German Credit* dataset. Ideal value of the former is 0.

Protected att.	**Baseline measures**			**Proposed global measures**		
	Statistical Parity	Equal Opportunity	Average Odds	Positive regions	Negative regions	Boundary regions
Age (young)	-0.28	-0.3	-0.25	0.01	0.01	0.04
Sex (female)	-0.002	0.04	-0.01	0.02	0.02	0.08

Conclusions. The proposed measures rely on an intuitive notion of explicit bias related to the uncertainty in decision-making as expressed by changes in the fuzzy-rough boundary regions. Our measures have several advantages that can be summarized as follows. First, the measures do not depend on any ML model. Second, the measures consider all features and feature-groups at once. This means that all available information is being leveraged and that arbitrary assumptions regarding the discriminated groups are avoided. Third, no discretization is needed during pre-processing to handle numeric features. Finally, the measures are not affected by data imbalances. Potential limitations of our approach include the limited number of considered literature measures, lack of experimentation with respect to bias that is implicitly encoded in non-sensitive features and dependence on the distance function and fuzzy operators.

As for the ramification of this thesis, we developed a stronger measure [9]. The corresponding paper received the Best Paper Award at the 25th Iberoamerican Congress on Pattern Recognition. An extended version of this work is currently under review for publication [10] at the Pattern Recognition Letters journal. Finally, we have recently submitted a journal contribution to the Neurocomputing journal where a neural model using a different approach confirmed the patterns found by the five proposed measures.

References

1. Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., Roth, D.: A comparative study of fairness-enhancing interventions in machine learning. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 329–338 (2019)
2. Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., Huq, A.: Algorithmic decision making and the cost of fairness. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 797–806 (2017)
3. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Computing Surveys (CSUR), **54**(6), 1–35 (2021)
4. Verma, S., Rubin, J.: Fairness definitions explained. In: 2018 IEEE/ACM International Workshop on Software Fairness (fairware), pp. 1–7 (2018)
5. Bellamy, R. K., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., ... Zhang, Y.: AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. IBM Journal of Research and Development, **63**(4/5), pp. 4–1. (2019)
6. Inuiguchi, M., Wu, W., Cornelis, C., Verbiest, N.: Fuzzy-Rough Hybridization. Handbook of Computational Intelligence (2015)
7. Pawlak, Z.: Rough sets. International Journal of Computer & Information sciences, **11**(5), pp. 341–356 (1982)
8. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. In: International Journal of General System, **17**(2-3), pp. 191–209 (1990)
9. Koutsoviti Koumeri, L., Nápoles, G.: Bias Quantification for Protected Featuresin Pattern Classification Problems, In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 25th Iberoamerican Congress, Lecture Notes in Computer Science, Springer (2021)
10. Nápoles, G., Koutsoviti Koumeri, L.: A fuzzy-rough uncertainty measure to discover bias encoded explicitly or implicitly in features of structured pattern classification datasets, arXiv (2021)

Learning Deep Coordination Graphs for Multi-Agent Systems

G. Wullaert, F. Perez Sanjines, T. Verstraeten, and A. Nowé

[1] Vrije Universiteit Brussel, Brussels, Belgium
[2] {gregory.nicolas.wullaert, timothy.verstraeten fabian.perez.sanjines ann.nowe}@vub.be

Autonomous systems have been essential for solving complex real-world problems and planning new efficiencies in various other domains. Reinforcement learning (RL) underlies how we model and think about autonomous decision-making systems. In RL, we model the decision-making system as an agent receiving observations from the environment and taking actions in that environment. The environment responds with a reward signal, and the objective is to maximize the total discounted reward. The agents usually achieves this objective by learning from feedback based on continuous trial and error.

In many scenarios, our agent may not be the only agent in the world, and there might be multiple agents trying to achieve their goal. This introduces many challenging problems. One of these problems is how multiple agents need to coordinate in order to maximize a shared reward function. A straightforward approach for coordination in a multi-agent setting is to reduce the problem statement to a single-agent reinforcement learning problem where the action space is the joint-action space of all agents in the environment. However, this approach becomes quickly infeasible due to the combinatorial increase of joint action-space in terms of the number of agents.

Many other exciting coordination methods can be found in the literature; however, the work done during my thesis focuses primarily on coordination graphs. In a coordination graph, each agent is represented as a node in the graph, and we are defining a set of edges between pairs of agents that corresponds to a payoff function that depends on the actions of both connected agents. This allows us to break down the coordination between all agents into a smaller coordination problem with fewer agents that is easier to solve.

In the literature, often a planning perspective is taken in order to create coordination graphs, rather than a learning perspective. This is due to the fact that creating these coordination graphs often requires domain expertise in their design. However, this may be challenging for dynamic environments with shifting coordination requirements.

We propose a novel sample-efficient reinforcement method based on deep implicit coordination graphs (DICG) [Li et al., 2021] to automatically learn the coordination graph between agents while learning the optimal joint control strategy. The deep implicit coordination graph infers the coordination graph using a self-attention network, which uses soft-edges to indicate the strength of the coordination between agents. Afterwards, it uses a graph neural network to learn the implicit relations about the joint-actions. We use this DICG to learn the

trade-off between fully centralized and decentralized learning via a soft-actor critic method, which is an off-policy reinforcement learning algorithm. Therefore, our method is most suitable for learning that takes place in a centralized environment, where we can share parameters, observations, gradients, and so forth., between all homogenous agents in the environment. However, we show that the learned policies by the agents can not only be executed in a centralized but also in a decentralized setting, where there is no communication between the agents.

Our approach is evaluated on the well-known Predator-Prey domain, where eight predators have to coordinate to catch the eight preys in the environment. We configure the environment to be a 10 x 10 grid world with a 5 x 5 grid view visibility for the predators with themselves at the center. If two predators capture the same prey, the predators receive a reward of 10. However, we penalize both predators with a negative reward if a single predator tries to capture the prey. Therefore, we require at least two predators to be present in the neighboring grid cells of prey to capture successfully. By introducing this negative reward, we show that our approach solves the relative overgeneralization pathology, i.e., other agents act randomly during exploration, and punishment caused by uncooperative agents may outweigh rewards achievable with coordinated actions.

Our approach proves to be more sample efficient and stable than previous approaches. This is because we are using off-policy reinforcement learning methods in contrast to the previous approach, where they used on-policy reinforcement methods.

The framework built in this work scales well for complex environments with changing dynamics, which means that our approach would perform well on real-world applications such as traffic light control, wind farm control, routing of taxi fleets, and drone swarms. Often, such settings can be formulated as coordination problems in which agents have to coordinate to optimize a shared team reward. In future work, we aim to specifically validate our method on a wind farm control case.

References

1. Li, S., Gupta, J. K., Morales, P., Allen, R., Kochenderfer, M. J. (2021). Deep implicit coordi-nation graphs for multi-agent reinforcement learning.arXiv preprint arXiv:2006.11438

Encoder-Decoder Approaches for Detection and Diagnosis of Anomalies in Machine Control Applications*

Julian Posch[1,2], Jacques Verriet[1], and Kurt Driessens[2]

[1] ESI (TNO)
[2] Data Science and Knowledge Engineering, Maastricht University

Faults and defects in machine control applications and industrial systems can have far-reaching and costly consequences in terms of downtime or damage to equipment. The time-series data resulting from such systems is high-dimensional and multi-modal in nature and the types of faults that might be encountered during operation are usually not fully known in advance [7]. Encoder-decoder architectures (e.g. Autoencoders, Transformers) using a reconstruction objective offer a framework in which only nominal system data is necessary to train models to detect and diagnose these faults as anomalies. These architectures learn to encode and subsequently decode (or reconstruct) nominal datapoints. They aim to detect anomalies by failing to accurately reconstruct them under the learned model, interpreting the reconstruction error as an anomaly score [4]. However, performing a case study showed that encoder-decoder architectures can prove difficult to control and interpret, posing risks for their reliability in practical applications. In response to the observed drawbacks we develop and empirically evaluate a novel architecture for anomaly detection, termed Self-Attention Autoencoder. Furthermore, an anomaly diagnosis methodology is proposed in order to assist machine engineers by identifying potential anomaly causes.

In order to develop an interpretable, multi-variate anomaly detector, the standard Transformer architecture [6] is adapted by removing all residual connections so as to achieve a straightforward flow of information. Next, with the intention of simplifying the model to its essentials, the decoder component of the model is completely removed. This means that input is now only fed into a single component and only a singular layer of attention remains. Furthermore, instead of Multi-Head Attention, Scaled-Dot-Product Attention [6] was used, which only produces a single attention matrix. The bottleneck and decoder of the new model are realized by converting the feedforward layer of the Transformer's encoder into an Undercomplete Autoencoder [3]. A visualization of the proposed architecture is shown in Figure 1. Once an anomaly is detected, potential anomaly causes are identified. The signal (variable) with maximal reconstruction error as well as attention matrix column with maximal attention are determined. Together, these establish where the model's reconstruction of the input was thrown off, giving a clear indicator for machine engineers to directly focus their anomaly cause investigation on.

* This research was carried out as part of the ITEA3 18030 MACHINAIDE project. Full text available at *https://github.com/JulianPosch/MSc-Thesis-Anomaly-DD*

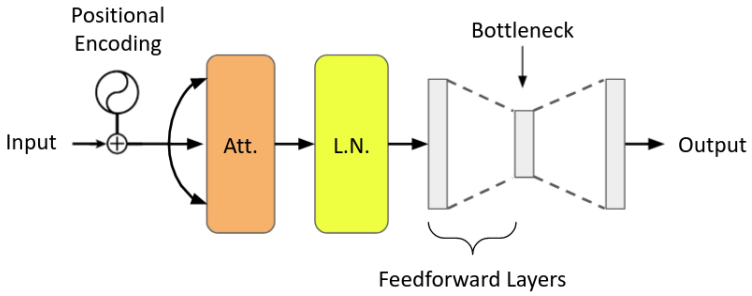


Fig. 1: Visualization of the proposed Self-Attention Autoencoder architecture. **Att.** and **L.N.** refer to Scaled Dot-Product Attention [6] and layer normalization [1] respectively.

The proposed Self-Attention Autoencoder is evaluated for its anomaly detection performance on digital twin data from an industrial machine setup. Performance is measured through AUC [2] and F1-Score [5] resulting from true and false positive rates at different anomaly score thresholds. The Self-Attention Autoencoder shows an AUC of 0.913 and F1-Score of 0.918, outperforming Undercomplete Autoencoders (AUC: 0.791, F1: 0.828), LSTM Autoencoders (AUC: 0.899, F1: 0.867) and matching the performance of Transformers (AUC: 0.917, F1: 0.905). Anomaly diagnosis performance is evaluated on synthetically generated data. This allows for control over the exact signal and timestep location of anomalies, making it possible to measure anomaly diagnosis performance quantitatively.[3] Both Transformer and Self-Attention Autoencoder perform similarly in terms of identifying anomalous signals among the multi-variate data, with an average of 19.5 and 19.6% of reconstruction error resulting from anomalous signals. When it comes to identifying the correct timestep of the anomaly however, a substantial difference is observed between the proposed Self-Attention Autoencoder and Transformer. The Self-Attention Autoencoder focuses its attention on timesteps corresponding to the anomaly in 55.9% of cases compared to the Transformer, which only does so in 17.9% of cases.

In conclusion, the architecture and methodology proposed in this thesis allow for both detecting anomalies as well as providing valuable information towards the spatial and temporal location of anomalies. The proposed Self-Attention Autoencoder matches the anomaly detection performance of the best evaluated encoder-decoder architecture (Transformer), while cutting down on complexity and including an easily controllable bottleneck. In terms of anomaly diagnosis performance, the Self-Attention Autoencoder vastly outperforms the Transformer, providing the potential for speeding up the efforts of engineers in resolving faults and defects.

[3] Since the proposed anomaly diagnosis methodology requires an attention mechanism, results will only be presented for the Transformer and Self-Attention Autoencoder.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization (2016)
2. Flach, P.: Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press (2017)
3. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), `http://www.deeplearningbook.org`
4. Ruff, L., Kauffmann, J., Vandermeulen, R., Montavon, G., Samek, W., Kloft, M., Dietterich, T., Müller, K.R.: A unifying review of deep and shallow anomaly detection. Proceedings of the IEEE **109**, 1–40 (02 2021)
5. Sammut, C., Webb, G.I. (eds.): Encyclopedia of Machine Learning. Springer US (2010)
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
7. Yin, S., Ding, S.X., Xie, X., Luo, H.: A review on basic data-driven approaches for industrial process monitoring. IEEE Transactions on Industrial Electronics **61**(11), 6418–6428 (2014)

Enhancing Reject Inference in Credit Scoring with Selective Semi-Supervised Learning

Anna-Maria Angelova, Fernando P. Santos, and Sandro Bjelogrlic

University of Amsterdam, The Netherlands

In banking industry, modelling the Probability of Default (PD) of loan applicants is a key component of credit risk management [7]. PD models are usually built on a sample of accepted borrowers and ignore the characteristics of rejected customers. As the accepted and rejected populations have different characteristics, a PD model may suffer from sample bias. Reject Inference (RI) refers to the techniques that try to remedy sample bias by inferring the performance of the rejected applicants. The goal of RI is to improve the performance of the PD model on the full through-the-door population (accepts and rejects).

Recent research suggests that RI can benefit from new modelling approaches inspired in machine learning, in particular semi-supervised learning (SSL). SSL-RI methods rely on labelled data from the accepted customers and unlabelled data from the rejected customers. There is evidence that SSL methods can outperform traditional RI approaches [2, 3, 5, 6], however, under certain conditions, SSL models can lead to undesirable outcomes [1, 4, 8–10]. These threats are part of the reason why traditional RI methods (e.g., simple augmentation [7]) are, still today, the most common approach in PD models.

In order to broaden the real-world application of SSL in RI, it is necessary to have a precise picture of the conditions that need to be met for this approach to be effective. We need conclusive evidence on the limits on unlabelled data that should be used in the SSL process. In this research, we investigate the quantity and type of data that are required for the successful application of selective SSL in RI. For that purpose, we systematically test the accuracy of SSL-RI varying 1) the percentage of unlabelled used in training and 2) the original distribution of applicants data. Based on edge-case data scenarios, we conclude that, if the default distributions of the rejected data are not significantly different than those in the accepted data and if an optimal range of (unlabelled) rejects are added to the training data, SSL-RI outperforms traditional methods.

Methodology and Results

We use as a baseline (to enhance with SSL and use as benchmark) a RI method commonly used in banking: simple augmentation [7]. In simple augmentation, a supervised classifier is trained on the accepted data and then used to score the rejects. A cutoff value is chosen to determine a classification threshold, above which the rejects will be classified as bad. The labelled accepted and the pseudo-labelled rejected data are then used to retrain the model. A disadvantage of conventional simple augmentation is that it is developed only on the labelled data. We propose to use selective SSL classifier (Self-Training) in place of the base supervised method with the goal of developing a model that can gradually

learn from the unlabelled rejected examples. During SSL training, the algorithm chooses the data-points with lowest and highest probabilities in the model – of being classified as good/bad – as the most confidently labelled examples that, as such, can be (pseudo-)labeled and added to an augmented training set. The experiment is conducted multiple times by adding different fractions of unlabelled data, while logging the performance on accepted and total samples.

To evaluate the effectiveness of the proposed approach, we perform two experiments, comparing simple augmentation with the SSL framework against the conventional simple augmentation approach. Experiment 1 demonstrates the variability of the results with respect to the amount of unlabelled data used in the training process. Experiment 2 tests the sensitivity of the outcome to different scenarios for default and reject rates. 16 scenarios in total are tested, including 4 different data distributions and 4 different combinations of default and reject rates. While the full results can be accessed in the thesis [1], here we present the optimal fraction of unlabeled data to be considered for two specific data distributions (see figure below). These two edge cases are chosen to represent a situation, where the labels in the rejected region could be inferred by extrapolation from the accepted region. Further tests are performed to assess the effectiveness of the RI-SSL method when the accepted and rejected data are drawn from significantly different distributions.

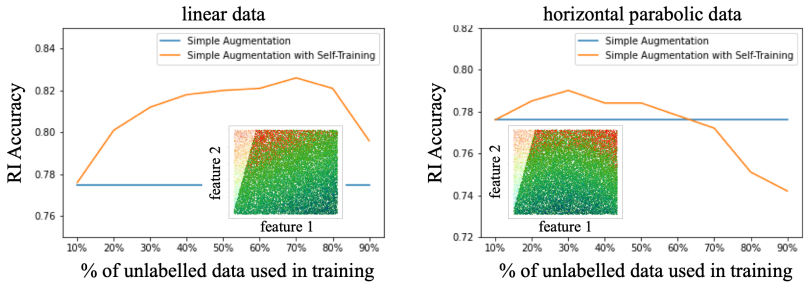


Fig. 1: We present RI accuracy for 2 data distributions (left, linear; right, horizontal parabolic) and for different percentages of unlabeled data used in training (horizontal axis). The inset figures represent the synthetic data distribution, where green is good (pays loan), red is rejected (defaults) and faded region corresponds to rejected data. In this example, we consider 10% of default and 10% of 10% rejected individuals.

The results of our experiments indicate that RI-SSL enhances the performance of simple augmentation under two conditions: First, the distribution of the unlabelled data should be similar to the distribution of the labelled data. Second, there is an optimal fraction of unlabelled data that should be added in the training process; adding all unlabelled examples on the through-the-door population is detrimental for RI accuracy.

[1] https://scripties.uba.uva.nl/search?id=722905

References

1. Chapelle, O., Schlkopf, B., Zien, A.: Semi-Supervised Learning. The MIT Press, 1st edn. (2010)
2. Kang, Y., Jia, N., Cui, R., Deng, J.: A graph-based semi-supervised reject inference framework considering imbalanced data distribution for consumer credit scoring. Applied Soft Computing **105**, 107259 (2021)
3. Kozodoi, N., Katsas, P., Lessmann, S., Moreira-Matias, L., Papakonstantinou, K.: Shallow self-learning for reject inference in credit scoring (09 2019)
4. Li, Y.F., Zhou, Z.H.: Towards making unlabeled data never hurt. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**, 175–188 (2015)
5. Li, Z., Tian, Y., Li, K., Zhou, F., Yang, W.: Reject inference in credit scoring using semi-supervised support vector machines. Expert Systems with Applications **74** (01 2017)
6. Maldonado, S., Paredes, G.: A semi-supervised approach for reject inference in credit scoring using svms. vol. 6171, pp. 558–571 (07 2010)
7. Naeem, S.: Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring (2005)
8. Oliver, A., Odena, A., Raffel, C., Cubuk, E.D., Goodfellow, I.J.: Realistic evaluation of deep semi-supervised learning algorithms. CoRR **abs/1804.09170** (2018)
9. Singh, A., Nowak, R., Zhu, J.: Unlabeled data: Now it helps, now it doesn't. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems. Curran Associates, Inc.
10. Zhu, X.: Semi-supervised learning literature survey. Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison

Online Learning of Deeper Variable Ordering Heuristics for Constraint Optimisation Problems

Floris Doolaard and Neil Yorke-Smith[0000−0002−1814−3515]
f.p.doolaard@student.tudelft.nl,n.yorke-smith@tudelft.nl

Delft University of Technology, The Netherlands

Abstract. Solvers for constraint optimisation problems exploit variable and value ordering heuristics. Numerous expert-designed heuristics exist, while recent research uses machine learning to learn novel heuristics. We introduce the concept of *deep heuristics*, a data-driven approach to learn extended versions of a given variable ordering heuristic. We demonstrate deep variable ordering heuristics based on the smallest, anti first-fail, and maximum regret heuristics. The results show that deep heuristics solve 20% more problem instances than classical 'shallow' heuristics.

1 Motivation and Approach

The order in which the variables are chosen can have significant effect on the total runtime of a constraint optimisation problem solver [3]. We address the situation of online solving of unseen optimisation problems. We introduce *deep variable ordering heuristics*, approximation functions that look at multiple levels of a search tree with the aim of generalizing better than classical heuristics.

As summarised in Figure 1, we implement deep heuristics in the open source Gecode solver [5]. Given a problem instance, an initial probing phase employs pseudo-random search to gather a variety of variable-value assignments. This data is then utilised by the machine learning component to acquire a deep heuristic function. Then second, during solving, given the current search state, the solver can predict scores with the learned model and select the variable with the best predicted score. Third, to leverage the pseudo-random nature of the probing data, a restart-based search strategy allows for multiple ML models to be learned, increasing the chance of finding solutions.

Chu and Stuckey [1] use online learning to acquire value heuristics: we learn variable ordering heuristics and we utilise a more complex score function. We use deeper lookaheads than Glankwamdee and Linderoth [4], and exploit ML predictions to circumnavigate the cost of lookaheads during search.

2 Results and Discussion

We test deep heuristics on four representative problem classes from the MiniZinc benchmarks: Resource Constrained Project Scheduling Problem (RCPSP), Evilshop, Amaze, Open Stacks. Instances are run for a maximum time of 4 hours.

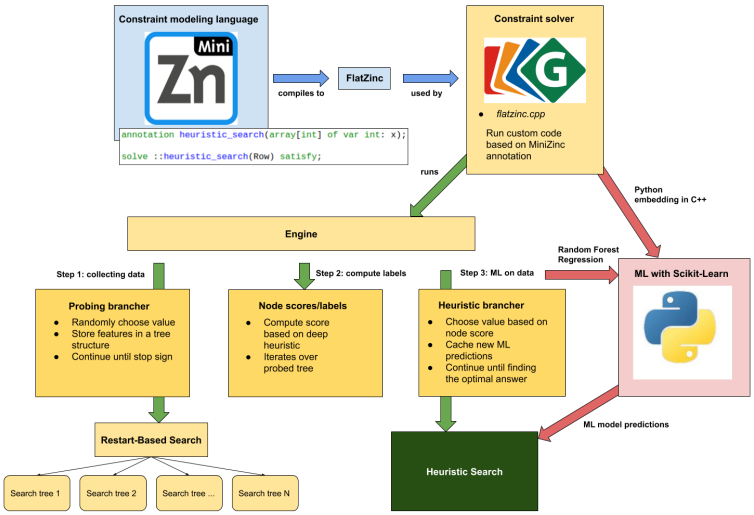
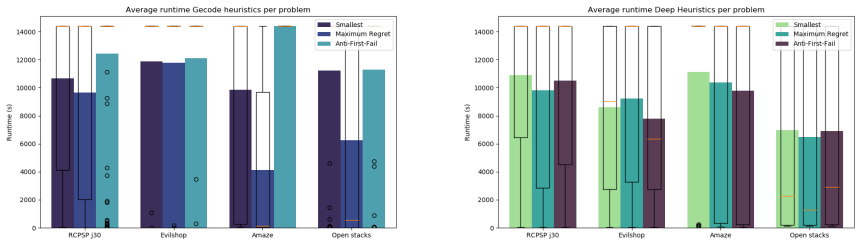


Fig. 1: Probing, learning, and heuristic search phases implemented in Gecode.



(a) Gecode heuristics (b) Deep heuristics

Fig. 2: Comparison of mean runtime between heuristics

Results, such as shown in Figure 2, indicate that the deep heuristics often – but not always – outperform the 'classical' version of the heuristics. For the deep heuristics, the runtime includes the probing and training time, as well as the solving time. Full results are found in the thesis [2]. Overall we find that deep heuristics solve 20% more problem instances, while improving on total runtime for the Open Stacks and Evilshop benchmark problems.

The thesis provides a novel approach to one-shot learning of search heuristics for constraint optimisation problems. Further experiments are warranted to assess the contribution of each the parts of our approach. In particular, recognising the stochasticity inherent in a learning-based approach, we use restarts with the deep heuristics – but not with their classical counterparts.

References

[1] Chu, G., Stuckey, P.J.: Learning value heuristics for constraint programming. In: Proceedings of the 12th International Conference on the Integration of AI and OR Techniques in Constraint Programming (CPAIOR'15), p. 108–123 (2015)
[2] Doolaard, F.: Deepification: Learning Variable Ordering Heuristics in Constraint Optimization Problems. Masters thesis, Delft University of Technology (2020)
[3] Gent, I.P., MacIntyre, E., Prosser, P., Smith, B.M., Walsh, T.: An empirical study of dynamic variable ordering heuristics for the constraint satisfaction problem. In: Proceedings of 2nd International Conference on Principles and Practice of Constraint Programming (CP'96), pp. 179–193 (1996)
[4] Glankwamdee, W., Linderoth, J.: Lookahead branching for mixed integer programming. Tech. rep., Lehigh University (Oct 2006)
[5] Schulte, C., Tack, G., Lagerkvist, M.Z.: Modeling and Programming with Gecode 6.2.0 (2019), URL www.gecode.org

Explaining the Behavior of Remote Robots to Humans (Extended abstract)*

Yazan Mualla[1], Stéphane Galland[1], and Christophe Nicolle[2]

[1] CIAD, Univ. Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France
[2] CIAD UMR 7533, Univ. Bourgogne Franche-Comté, UB, F-21000 Dijon, France

In the future AI systems, it is vital to guarantee a smooth human-agent interaction, and explainability is an indispensable ingredient for such interaction [16, 17]. Accordingly, the research domain of Explainable AI (XAI) is gaining increased attention from researchers of various disciplines [4, 5, 1]. When providing explanations to humans, the aim is to imitate how they generate and communicate everyday explanations in their everyday life [7]. This leads us to discuss the *parsimony of explanations* [6, 15] that could help in providing the necessary information while reducing the *human cognitive load* to avoid overwhelming the human with useless information [18], *i.e.* there is a trade-off between the two features of an explanation, namely *simplicity* and *adequacy* [2, 3].

The thesis presents a mechanism for parsimonious XAI that strikes a balance between simplicity and adequacy. In particular, it introduces a *context-aware* and *adaptive* process of explanation formulation and proposes a human-agent architecture allowing to make this process operational for remote robots represented as Belief-Desire-Intention (BDI) agents [14]. To formulate parsimonious explanations, the proposed architecture relies first on generating *normal* explanations, in relatively normal situations, and *contrastive* ones [8] in abnormal situations; second on updating and filtering the explanations before communicating them to the human. The architecture investigates the three phases of providing an explanation from agents to the human: generation, communication, and reception [13]. We argue that a well-formed combination of these phases leads to formulating a parsimonious explanation.

The research methodology is five-fold (Figure 1): ***(a)*** Identify open research issues after performing a Systematic Literature Review (SLR) [10]. ***(b)*** Define Research Questions (RQs) based on the identified research issues: **RQ1)** Does explainability increase the humans' understandability of the remote robots represented as agents? **RQ2)** How to strike a balance between simplicity and adequacy? **RQ3)** Are the cognitive architecture and the BDI model good candidates for human-agent explainability? ***(c)*** Structure the RQs in Research Hypotheses (RHs) that can be statistically analyzed[3]. ***(d)*** Propose the architecture, model, and process to answer the RQs. ***(e)*** Conduct a specific experimental methodology to evaluate the proposals by statistically investigating the RHs according to the recommendations in the XAI domain.

* This work is a synthesis of the Ph.D. thesis defended at Université Bourgogne Franche-Comté, France on the 30th of November 2020 [9].
[3] Details about the RQs and the RHs can be found in the thesis [9].

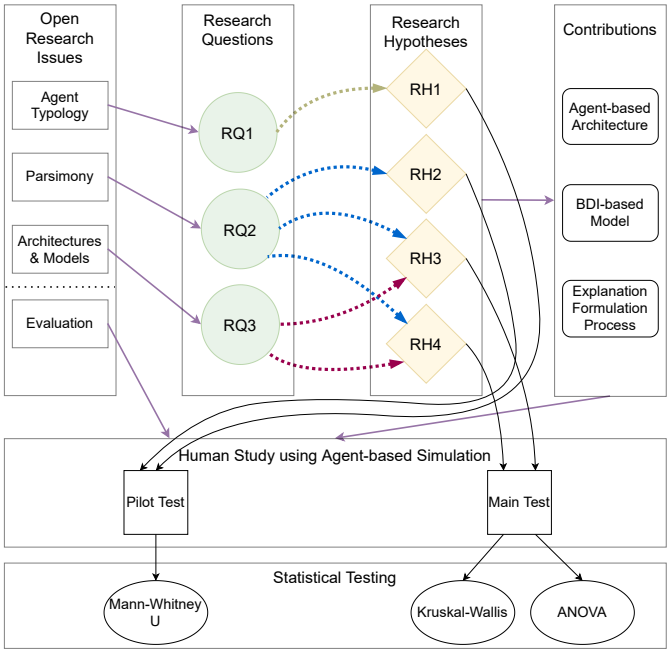


Fig. 1. Research methodology of the thesis.

The human understandability of AI is subjective, and this emphasizes the importance of empirical human user studies, where the users' opinions on the usefulness of explanations are investigated [8]. We have conducted two tests: Pilot test (*Mann-Whitney U*) [12] and Main test (*Kruskal-Wallis* and *ANOVA*) [11]. The responses of the participants (or users) are statistically analyzed and validated in terms of significance for both tests.

The pilot test discusses RQ1 and its results show that the explanation increases the ability of users to understand the explanations of remote robots. However, too many details overwhelm the users; hence, the filtering of explanations, that provides less, concise, and synthetic explanations, is preferable. In the main test, RQ2 and RQ3 are handled. It is proved that a combination of the phases of explanation generation and communication is needed to formulate the most useful explanation for the user. Comparing several combinations of parsimonious explanation formulation, it is proven that the best one includes using adaptive filtering with both normal and contrastive explanations. Regarding RQ3, the results revealed that the BDI model helps in realizing the explanation formulation process, as it organizes the various interactions between the system entities and allows for an adaptive and context-aware response based on the changes in the beliefs and intentions of the agents. Future work could investigate more the *human-centered* or *user-aware* XAI approaches and the verification and validation of XAI Systems.

References

1. Calvaresi, D., Mualla, Y., Najjar, A., Galland, S., Schumacher, M.: Explainable multi-agent systems through blockchain technology. In: Calvaresi, D., Najjar, A., Schumacher, M., Främling, K. (eds.) Explainable, Transparent Autonomous Agents and Multi-Agent Systems. pp. 41–58. Springer International Publishing, Cham (2019)
2. Chomsky, N., Collins, C.: Beyond explanatory adequacy, vol. 20. mitwpl (2001)
3. Contreras, H.: Simplicity, descriptive adequacy, and binary features. Language pp. 1–8 (1969)
4. Gunning, D.: Explainable artificial intelligence (XAI). Defense Advanced Research Projects Agency (DARPA), nd Web (2017)
5. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: XAI—Explainable Artificial Intelligence. Science Robotics (2019)
6. Laird, J.: The law of parsimony. The Monist **29**(3), 321–344 (1919), http://www.jstor.org/stable/27900747
7. Malle, B.F.: How the mind explains behavior: Folk explanations, meaning, and social interaction. Mit Press (2006)
8. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence **267**, 1–38 (2019)
9. Mualla, Y.: Explaining the Behavior of Remote Robots to Humans : An Agent-based Approach. Theses, Université Bourgogne Franche-Comté (Nov 2020), https://tel.archives-ouvertes.fr/tel-03162833
10. Mualla, Y., Najjar, A., Daoud, A., Galland, S., Nicolle, C., Yasar, A.U.H., Shakshuki, E.: Agent-based simulation of unmanned aerial vehicles in civilian applications: A systematic literature review and research directions. Future Generation Computer Systems **100**, 344–364 (2019)
11. Mualla, Y., Tchappi, I., Kampik, T., Najjar, A., Calvaresi, D., Abbas-Turki, A., Galland, S., Nicolle, C.: The quest of parsimonious xai: a human-agent architecture for explanation formulation. Artificial Intelligence p. 103573 (2021)
12. Mualla., Y., Tchappi., I., Najjar., A., Kampik., T., Galland., S., Nicolle., C.: Human-agent explainability: An experimental case study on the filtering of explanations. In: Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 1: HAMT,. pp. 378–385. INSTICC, SciTePress (2020). https://doi.org/10.5220/0009382903780385
13. Neerincx, M.A., van der Waa, J., Kaptein, F., van Diggelen, J.: Using perceptual and cognitive explanations for enhanced human-agent team performance. In: International Conference on Engineering Psychology and Cognitive Ergonomics. pp. 204–214. Springer (2018)
14. Rao, A.S., Georgeff, M.P., et al.: Bdi agents: from theory to practice. In: ICMAS. vol. 95, pp. 312–319 (1995)
15. Rasmussen, C.E., Ghahramani, Z.: Occam's razor. In: Advances in neural information processing systems. pp. 294–300 (2001)
16. Rosenfeld, A., Richardson, A.: Explainability in human–agent systems. Autonomous Agents and Multi-Agent Systems pp. 1–33 (2019)
17. Sokol, K., Flach, P.: Desiderata for interpretability: explaining decision tree predictions with counterfactuals. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 10035–10036 (2019)
18. Sweller, J.: Cognitive load theory. In: Psychology of learning and motivation, vol. 55, pp. 37–76 (2011)

Identifying strong predictors of engagement in Facebook news posts*

Pietro Piccini

Maastricht University, Maastricht, The Netherlands.

Abstract. In this paper, a data-set from Facebook news posts is constructed in order to measure the engagement of users with different news items. Logistic Regression is used as a baseline classifier to identify important post characteristics (e.g. topic, Page type, posting time, etc.) with respect to engagement.

Keywords: Social media · User engagement · News

1 Background & Data

In recent years social media has become a major news source for an increasing number of people and the task of modelling user engagement behaviour is becoming very relevant. This research aims at extending the current literature and offering new insights. The initial data was collected through the use of Crowdtangle, which is a platform that allows third parties to collect social media data. From there, some features are extracted to enhance the data-set. One of the most important extracted features is the topic of a post which is extracted using NLP techniques. Another important change to the data-set was the construction of an engagement metric that could describe the engagement level of a post with respect to how big, in terms of followers, the page that posted the news is. The solution for this problem was found by taking the ratio between total interactions and number of followers. The resulting feature which will be referred as the "total interactions ratio" solved some of the problem encountered when using the overperforming score calculated by Crowdtangle, mainly, the fact that it had a very unbalanced distribution, resulting in an unbalanced model. The full pipeline is described in figure 1.

2 Analysis results and Discussion

In order to analyze user engagement patterns I have used a binary classification model that takes all of the post features as argument and the binarized total interaction score (TIR) as target variable. The result is a Logistic Regression

* This thesis was prepared in partial fulfilment of the requirements for the Degree of Bachelor of Science in Data Science and Artificial Intelligence, Maastricht University. Supervisor: Jerry Spanakis

Fig. 1. Data-model pipeline

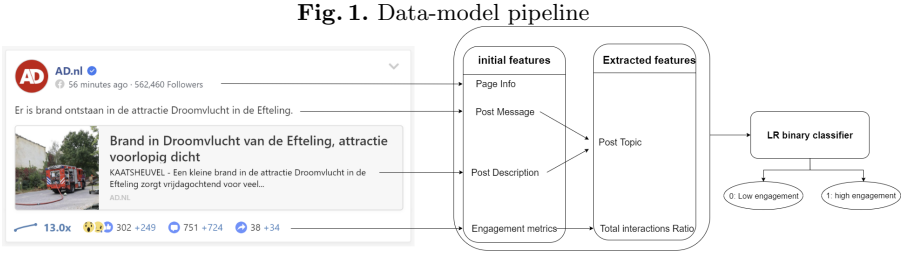


Table 1. Model F1-scores per class of TIR classification model

	F1-Score	
	Low Engagement	***High Engagement***
Logistic Regression	66.8%	66.4%
Ridge Classifier	66.6%	66.7%
Neural Network	68.3%	67.9%

model with a performance described in table 1 together with other tested algorithms. To identify strong predictors of engagement I looked at the model's coefficients which showed that the most influential variable for user engagement is the category of the page, followed by the page name, post type, topic, and finally, date and time. The coronavirus topic was by far the most correlated with high engagement, furthermore, it showed a frequency of angry and sad reactions that was much higher than the average. The page category that brings the most engagement was radio stations. The coefficients from the time of posting showed that the best possible time for posting news on Facebook is the evening with a sharp decline during night hours. In conclusion, the performance of the model is in line with models proposed in previous publications but its analysis brought new insights into user engagement patterns. Further research is required to deepen the analysis and gain a better interpretation.

References

1. Sotiris Lamprinidis, Daniel Hardt, and Dirk Hovy. Predicting news head-line popularity with syntactic and semantic knowledge using multi-tasklearning. InProceedings of the 2018 Conference on Empirical Methodsin Natural Language Processing, pages 659–664, Brussels, Belgium,October-November 2018.
2. Kholoud Khalil Aldous, Jisun An, and Bernard J. Jansen.: Predictingaudience engagement across social media platforms in the news domain. In: Ingmar Weber, Kareem M. Darwish, Claudia Wagner, Emilio Zagheni,Laura Nelson, Samin Aref, and Fabian Fl ock, editors, Social Informatics,pages 173–187, Cham, 2019. Springer, Heidelberg (2016).
3. Katuwal Rakesh and Ponnuthurai N Suganthan.An ensemble ofkernel ridge regression for multi-class classification.Procedia ComputerScience, 108:375–383, 2017.

Producing "Open-Style" Choreography for K-Pop Music with Deep Learning*

Songha Ban[1] and L.L. Sharon Ong[1]

Department of CSAI, Tilburg University, Tilburg, the Netherlands
{s.ban,l.l.ong}@tilburguniversity.edu

Abstract. The goal of this thesis to generate natural and beat-matching choreography from music using deep learning. We compare different pose estimators to create a dataset of human figures to generate dance. Our framework comprises of a music encoder to create music features which is fed to a pose generator to create dance pose sequences, as well as music feature generator which reconstructs music features from output poses to improve music feature encoding. Our results showed a pose estimator with a GRU music encoder, generated more natural dance movements which match K-Pop music compared to previous work.

Keywords: Dance Choreography Generation · Pose Estimation · Music Encoders

1 Introduction

Dance and music are abstract art forms. There are no established rules relating dance to music [3] particularly for the open-style or urban dances, which is the focus of our work. This style is creative and expressive, comprising of spontaneous body movements inspired by dance genres such as street, hip-hop, contemporary and jazz. The goal of this thesis is to generate natural open-style sequences of dance poses from K-Pop music using generative techniques. It is more complex to design and create aesthetic and rhythmic choreography with this music as it requires creative diverse dance techniques and comprehension of musical elements.

Previous work [5] explored generative techniques trained on different styles to create dance movements with human figures from music. Dancers specialize in a few styles, hence training with different styles could have resulted in dances with style-inconsistencies. This thesis extends the work in [5] with a music feature generator, inspired by the idea that humans can infer some audio features such as beat and tempo from watching dance. Hence, good dance movements should be able to generate music features. We trained on a dataset focusing on open-style dances to K-pop music. The choreography generated more natural open-style dance poses for the music. Successful choreography generation would benefit applications such as robotics, gaming, animation, and virtual reality.

* Source code available at `https://github.com/SonghaBan/DancingAI`. A demo video of the experiments is available at `https://youtu.be/UE9QnT59LlI`

2 Methods

The dataset consisted of 50 dance videos (115986 frames in total with a resolution of 640 x 320 and 30 fps) provided by Nataraja Academy available on YouTube and the first author. Pose estimation methods were applied on the raw dance videos to extract key points on the dancer's body, to build a dataset for our deep learning framework. A pose cleaning method filtered out poses in one frame which deviate significantly from previous frame. These poses were recovered using spline interpolation.

Our goal is to generate dance choreography by learning a model $G : X \longrightarrow Y$, where X is the music input and Y is the set of dance poses such that the distribution of the generated choreography $G(X)$ is undistinguisable from the distribution of the real dance poses Y. The original framework proposed in [5] comprised of a music encoder and a pose generator. The music encoder transforms the audio input into a hidden sequence of music features. In the thesis, we compared two models for music encoding (LTSM and GRU). The pose generator generated the poses Y which made up the choreography. This thesis extended the original framework [5] with an addition of a music feature generator. The music feature generator regenerated the music features from the generated pose sequences. A Local Temporal Discriminator evaluated our model on the coherence of consecutive frames. A Global Content Discriminator used self-attention mechanism [4] to obtain a comprehensive embedding and classified whether the pose sequence matched the music features.

3 Results

The Frechet Inception Distance (FID) [2] was used to assess the distance similarity between the generated dances and the real dances. The beat coverage ($\frac{\#motion\ beats}{\#music\ beats}$) and beat hit rate ($\frac{\#aligned\ beats}{\#motion\ beats}$) were evaluated. Music beats were obtained by computing onset strength from the audio [1]. Motion beats were detected by using standard deviation [6]. Our framework with LSTM in the music encoder performed best in beat coverage and slightly worse in FID and beat hit rate than the original framework trained on our dataset. The new framework with GRU in the music encoder resulted in the best in FID and beat hit rate. Our results were also evaluated by real dancers, which prefered our model compared to previous work.

Table 1. Result of quantitative evaluation.

Method	FID	Beat Coverage (%)	Beat Hit Rate (%)
Ren et al.	18.3	47.9	89.9
Original	16.6	50.0	91.2
LSTM	16.0	**51.3**	90.6
GRU (my model)	**14.8**	50.4	**91.5**

References

1. Ellis, D.P.: Beat tracking by dynamic programming. Journal of New Music Research **36**(1), 51–60 (2007). https://doi.org/10.1080/09298210701653344
2. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. Advances in Neural Information Processing Systems **2017-Decem**(June 2019), 6627–6638 (2017)
3. Lee, H.Y., Yang, X., Liu, M.Y., Wang, T.C., Lu, Y.D., Yang, M.H., Kautz, J.: Dancing to music. arXiv (NeurIPS), 1–11 (2019). https://doi.org/10.4135/9781446251409.n4
4. Lin, Z., Feng, M., Dos Santos, C.N., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. ICLR (2017)
5. Ren, X., Li, H., Huang, Z., Chen, Q.: Self-supervised Dance Video Synthesis Conditioned on Music. Proceedings of the 28th ACM International Conference on Multimedia pp. 46–54 (2020). https://doi.org/10.1145/3394171.3413932
6. Yalta, N., Watanabe, S., Nakadai, K., Ogata, T.: Weakly-Supervised Deep Recurrent Neural Networks for Basic Dance Step Generation. Proceedings of the International Joint Conference on Neural Networks **2019-July**(June 2020), 1–8 (2019). https://doi.org/10.1109/IJCNN.2019.8851872

Fine-Tuning Pretrained Language Models for Controlled Text Generation with Adapters

Valerie S. Sawirja and Peter Bloem

Vrije Universiteit, Amsterdam, The Netherlands
v.s.sawirja@gmail.com, p.bloem@vu.nl

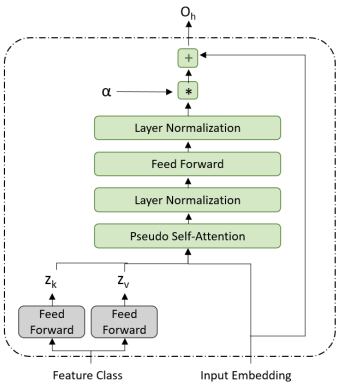
Auto-regressive language models (LM), such as GPT-2 [7], are pretrained into non-conditional models with the ability to generate realistic text. Still, their capability to produce text in a specified style is rather limited. Desired features, such as sentiments, cannot directly be included.

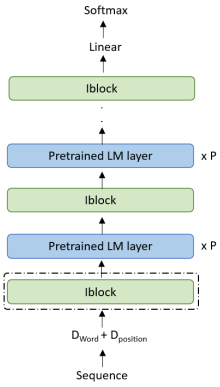
One method to incorporate this control in text generation is with conditional fine-tuning. Earlier works use pseudo self-attention (PSA) in the transformer layers of a pretrained GPT-2 [1, 3, 11]. PSA lets the representation of a desired feature precede all the input tokens for certain elements of the transformer self-attention [11]. This enables the desired feature to be directly absorbed into the output of the transformer layers, thus conditioning the model output. However, these past studies use an all-layer fine-tuning method, which requires many resources. Using this has been criticized for its financial and ecological affects [9].

An alternative is fine-tuning with adapters. These are modules with task-specific parameters [8, 2]. Adapters are randomly initialized and injected to augment a pretrained LM architecture, while freezing the pretrained weights. These new parameters are then trained to adapt the LM to a downstream task. Only a handful of parameters have to be trained, making it less costly to fine-tune an LM. The resulting fine-tuned LMs can moreover be re-purposed by only exchanging the adapter modules [6]. With the current usage consensus, every adapter in a pretrained layer is mapped onto one specific task [4, 6]. To enable the conditional fine-tuning of an LM, we focus on an alternate adapter configuration.

Our adapter, called an iblock, is a transformer layer that is composed of PSA [11] and a feed forward, both followed by a layer normalization. The iblocks have a coefficient on a residual connection that is initialized at zero [8], which sets the pretrained LM as the starting point for fine-tuning. Different from [8], we include a dropout and a cap as optional hyper-parameters for this coefficient. The linear parameters of the last layer normalization have to be disabled for the regularization methods to work. The iblocks are placed on an interval through the pretrained architecture. These configurations are visualized in Figure 1. In our experiments, we use a 12-layer pretrained GPT-2 and 4 total iblocks.

We first investigate whether a pretrained GPT-2 could be conditionally fine-tuned. We use the IMDb dataset for sentiment [5] and AG News for topics [10]. For both we observe that the perplexity decreases after conditional iblock fine-tuning, compared to the pretrained GPT-2. The coefficients appear to be an important model element, as disabling them increases the perplexity on both datasets. We moreover find that regulating the coefficient values with a dropout and a value cap can limit catastrophic forgetting to some degree. Applying a cap





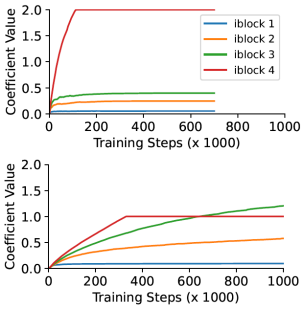


Fig. 1. The model architecture, with the internals of one iblock (left) and the placing of the iblocks within a pretrained GPT-2 (right), with z as the feature representation, α as the residual connection coefficient, D representing an embedding, and P portraying an interval of pretrained layers between iblocks.

Fig. 2. The coefficient development during conditional fine-tuning on IMDb (top) and AG News (bottom), where iblock 1 is the lowest adapter in the architecture.

on the top iblock allows the model to converge on IMDb. For AG News, using a cap decreases the level of forgetting, but does not prevent it completely. The development of the coefficients during fine-tuning is displayed in Figure 2.

Since successful conditioning is unclear from merely the perplexity, we perform a text generation experiment. We use a conditional and a non-conditional GPT-2, both fine-tuned on the IMDb data. We generate 400 total samples: 200 non-conditional samples and 100 samples per conditional feature class. These are blindly annotated with either a Positive, Negative, or Unclear sentiment. 35.50% of the conditional samples are annotated with Unclear, compared to 51.50% of the non-conditional samples. The accuracy of the conditional samples is higher for the positive samples (60%) than the negative samples (45%), but neither is significant. These results indicate an effect of conditioning, although the accuracy itself is not significant with our sample size.

In short, we investigate the application of residual transformer adapters with extra features for controlled text generation. We implement this for the conditional fine-tuning of a pretrained, auto-regressive, non-conditional LM and identify a partial success. We discover that it is valuable to include a coefficient on the adapter's residual connection, both as a learned parameter and as an element to regularize against catastrophic forgetting. Still, more research is needed with regards to the generation accuracy of the fine-tuned models. A further limitation of this study is the absence of a baseline model comparison.

Overall, with the used datasets and model configurations, training the transformer adapters requires less than 24 hours. With future studies, they may be able to conditionally fine-tune large scale transformer language models in a cost-effective manner.

References

1. Fang, L., Zeng, T., Liu, C., Bo, L., Dong, W., Chen, C.: Transformer-based conditional variational autoencoder for controllable story generation. arXiv preprint arXiv:2101.00828 (2021)
2. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: International Conference on Machine Learning. pp. 2790–2799. PMLR (2019)
3. Li, C., Gao, X., Li, Y., Peng, B., Li, X., Zhang, Y., Gao, J.: Optimus: Organizing sentences via pre-trained modeling of a latent space. arXiv preprint arXiv:2004.04092 (2020)
4. Lin, Z., Madotto, A., Fung, P.: Exploring versatile generative language model via parameter-efficient transfer learning. arXiv preprint arXiv:2004.03829 (2020)
5. Maas, A., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies. pp. 142–150 (2011)
6. Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., Cho, K., Gurevych, I.: Adapterhub: A framework for adapting transformers. arXiv preprint arXiv:2007.07779 (2020)
7. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
8. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. arXiv preprint arXiv:1705.08045 (2017)
9. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in nlp. arXiv preprint arXiv:1906.02243 (2019)
10. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. Advances in neural information processing systems **28**, 649–657 (2015)
11. Ziegler, Z.M., Melas-Kyriazi, L., Gehrmann, S., Rush, A.M.: Encoder-agnostic adaptation for conditional language generation. arXiv preprint arXiv:1908.06938 (2019)

Explainable Reinforcement Learning for Fleet Applications

Thomas Vaeyens, Youri Coppens, Timothy Verstraeten, Ann Nowé

Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels
thomas.lieven.vaeyens@vub.be

Reinforcement Learning (RL) has become an important driver to tackle complex control problems in a variety of applications. Specifically, in the context of fleet control, RL methods are used to control a collection of similar, but non-identical, machines that are performing the same task. Due to these similarities, fleet members can exchange knowledge with each other in order to improve sample-efficiency of the learning process.

However, the training process and the resulting control policies of current RL methods are challenging to explain and justify to humans. In many high-risk domains, the lack of interpretability and transparency makes it difficult, if not dangerous, to trust the output of such models.

We propose a novel explainable fleet-wide control learning algorithm. Specifically, a coregionalized Gaussian process is used to capture the transition model of multiple members in the fleet, after which a policy iteration method is applied to optimize the control policy of each specific member [4]. Afterwards, these policies can be distilled into a more compact and explainable Soft Decision Tree (SDT) model [1], which is a binary decision tree with weighted edges that reflect the importance of control decisions given the observed state information.

Furthermore, we extended the method with two transfer learning methods, i.e. TrAdaBoost [2], a boosting method for a fleet member's model that weights the importance of the data of another fleet member, and Multi-class TrAdaBoost [3], an extension that uses a multi-class exponential over the set of possible actions. The introduction of transfer learning further leverages the similarities between fleet members in order to share data. This improves the sample-efficiency while learning an explainable policy. Specifically, fleet members that are similar enough should transfer training data between each other. The decision which pair of members are similar is based on the average correlations between those members learned by the Gaussian process.

An extensive analysis was performed to compare the different methods on fleet variants of two classic control benchmarks: mountain car and cart pole. For each experiment, a fleet consists of one target member, a source member similar to the target, and different source member. The differences between the members are simulated by varying the power of the cars and mass of the poles. The goal of the target is to learn an efficient control policy, while leveraging the data obtained by the other two source members. Based on the obtained fleet policy, five different SDT models are distilled. Specifically, two baseline models are trained, one for the target and one for the source, using the traditional training method with a large data set. Moreover, three different models are

trained for the source: one using the standard TrAdaBoost algorithm, one using the standard TrAdaBoost algorithm for a single iteration with the correlation between the target and the source as fixed weight, and one using the Multi-class TrAdaBoost algorithm. For these last three models, a large data set from the source and a small data set from the target was used.

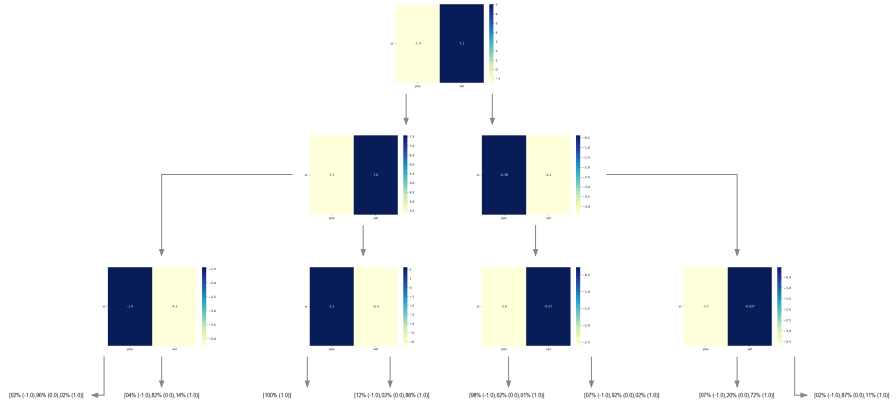


Fig. 1. SDT model visualisation with depth three for source (SA) from the mountain car fleet obtained with the Multi-class TrAdaBoost algorithm, with weight updates for 100 iterations using the large dataset from target (T) and the smaller dataset from the source (SA).

The results were analysed using the obtained SDT visualisations as shown in Figure 1 by comparing the baseline SDT models trained with the standard training method to the SDT models obtained with the transfer learning methods in terms of similarity and performance. The correlation between the fleet members seems to play an important role in the explainability. With lower correlations, the difference between policies is bigger which reduces the effectiveness of the transfer learning to obtain a similar SDT model. Multi-class TrAdaBoost algorithm provides the best result in terms of accuracy and interpretability and provides an SDT model that is almost identical to the baseline SDT model. The standard TrAdaBoost algorithm also provides similar SDT models. The learning method in which we use the correlation as the initial weight for one iteration gives less desirable results in terms of interpretability. There is a trade-off that needs to be made between interpretability and accuracy, deeper trees collect more reward because they allow you to create more specific action distributions in the leaf nodes. This complexity comes with the downside of decreased interpretability, for this reason, we selected trees of depth three.

References

1. Youri Coppens, Kyriakos Efthymiadis, Tom Lenaerts, and Ann Nowé. Distilling Deep Reinforcement Learning Policies in Soft Decision Trees. In Tim Miller, Rosina Weber, and Daniele Magazzeni, editors, *Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence*, pages 1–6, 2019.
2. Wenyuan Dai, Qiang Yang, Gui Rong Xue, and Yong Yu. Boosting for transfer learning. In *ACM International Conference Proceeding Series*, volume 227, pages 193–200, 2007.
3. Hanxian He, Kourosh Khoshelham, and Clive Fraser. A multiclass tradaboost transfer learning algorithm for the classification of mobile lidar data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:118–127, 2020.
4. Timothy Verstraeten, Pieter J.K. Libin, and Ann Nowé. Fleet control using coregionalized gaussian process policy iteration. In *Frontiers in Artificial Intelligence and Applications*, volume 325, pages 1571–1578, 2020.

Bayesian Inverse Reinforcement Learning for strategy extraction in the iterated Prisoner's Dilemma game

Matthias Cami[1], *Supervisors*: Inês Terrucha[1,2], Yara Khaluf[3], and Pieter Simoens[1]

[1] Dept. of Information Technology - IDLab, Ghent University - imec, Belgium
[2] AI Lab, Vrije Universiteit Brussel, 1050 Brussels, Belgium
[3] Information Technology Group, Wageningen University and Research, The Netherlands

1 Introduction and Methodology

As Artificial Intelligence (AI) becomes more relevant in various fields, interactions between humans and artificial agents will become more and more common. In order to be trustworthy and acceptable as assistive agents, such agents should be able to predict and account for human preferences. In strategic situations as modelled in game theory, humans will often deviate from predicted equilibrium models. While it has been shown that AI agents can be trained to reach superhuman performance in zero-sum games like poker, their learned policies do not reflect typical human strategies and therefore are not suited to predict the actions of humans. However, this does not mean that humans act in an unpredictable manner, they follow their own preferences, that take into account both their opponent's actions and the context of the interaction. While a big part of AI research has been focusing on beating the opponent in zero-sum games, most interactions are actually "mixed motive", which means that the interests of the players are not completely aligned, but also not solely competitive. Since the goal of AI is to aid human decision-making, the problem then becomes: how can AI optimize for human social preferences that are not easily hard-coded but change according to different parameters?

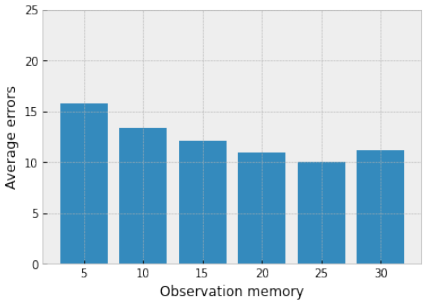
To tackle this question we use a classical mixed-motive game: the Prisoner's Dilemma (PD). Specifically, we focus on repeated interactions as they provide more diverse insights on how human preferences might change in accordance to the actions of the opponent. For this purpose we will use empirical data from the two-player iterated PD to illustrate the aforementioned dynamics [3]. To help AI infer human preferences in such setting we turn to Imitation Learning techniques, specifically the Bayesian Inverse Reinforcement Learning (BIRL) method [7]. As in Inverse Reinforcement Learning (IRL) [6], BIRL extracts the reward function from any given set of demonstrations with the added value that BIRL takes a probabilistic view of the reward. This means that, with BIRL, we can incorporate domain knowledge to choose the prior that selects from the (infinitely) many possible rewards for which the observed actions would be optimal.

2 Results and Discussion

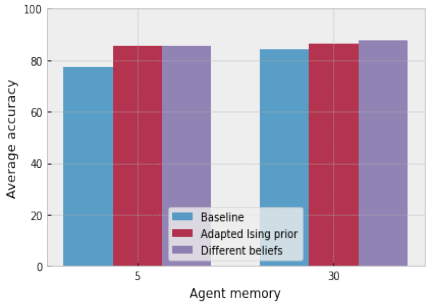
First, we tested whether the rewards humans follow are stationary. For this purpose we tested agents with different memory capacity about previous rounds of play. In Fig. a we show the results: it is clear that higher memory agents outperform the lower memory agents. Even larger memories were tested, but showed a decline in performance, suggesting that general human behavior is only stationary for a certain time frame.

To incorporate domain knowledge, in opposition to keeping the Uniform prior used in the baseline, we used an adapted version of the Ising prior [1]. We chose that for two reasons: to give more relevance to the more recent rounds when predicting the next [2, 5] and to emphasize a clear choice between the available actions by the expert. The results are seen in Fig. b where the 5-memory agent has the most significant increase in performance. This shows that a well constructed prior, using domain knowledge, can significantly help agents in their performance, even when there is not much data available to them.

The transition probabilities in the baseline model, which define the beliefs of the expert about their opponent's action, assumed equal chance of either action. To test different transition probabilities, we follow the principles of theory of the mind [4] and test whether assuming that the expert is able to correctly predict their opponent every period will influence accuracy. From the results in Fig. b we see that the accuracy increases very slightly, suggesting that even though perfect prediction is not realistic, randomness seems to perform worse.



(a) Average amount of errors for the baseline agent comparing different observation memories



(b) Average accuracy comparing 3 different agent setups on agents with memory 5 and 30

In conclusion we show that incorporating domain knowledge and probing the expert's beliefs increase the accuracy of the imitation technique used. The first proving to greatly increase the performance of the agent even for situations with small amount of available data (lower memory setups).

Acknowledgements

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

1. Cipra, B.A.: An introduction to the ising model. The American Mathematical Monthly **94**(10), 937–959 (1987)
2. Gracia-Lázaro, C., Ferrer, A., Ruiz, G., Tarancón, A., Cuesta, J.A., Sánchez, A., Moreno, Y.: Heterogeneous networks do not promote cooperation when humans play a prisoner's dilemma. Proceedings of the National Academy of Sciences **109**(32), 12922–12926 (2012)
3. Grujić, J., Eke, B., Cabrales, A., Cuesta, J.A., Sánchez, A.: Three is a crowd in iterated prisoner's dilemmas: experimental evidence on reciprocal behavior. Scientific reports **2**(1), 1–7 (2012)
4. Jara-Ettinger, J.: Theory of mind as inverse reinforcement learning. Current Opinion in Behavioral Sciences **29**, 105–110 (2019)
5. Nay, J.J., Vorobeychik, Y.: Predicting human cooperation. PloS one **11**(5), e0155656 (2016)
6. Ng, A.Y., Russell, S.J., et al.: Algorithms for inverse reinforcement learning. In: Icml. vol. 1, p. 2 (2000)
7. Ramachandran, D., Amir, E.: Bayesian inverse reinforcement learning. In: IJCAI. vol. 7, pp. 2586–2591 (2007)

Clinical Predictive Models: A comparison between Machine Learning and Classical Techniques

Michela Venturini[1,2][0000−0002−9947−0218] and Giulia Barbati[3][0000−0001−8942−5686]

[1] KU Leuven, Campus KULAK - Department of Public Health and Primary Care, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
[2] ITEC - IMEC, Etienne Sabbelaan 51, 8500 Kortrijk, Belgium
[3] University of Trieste, Department of Medical Sciences - Biostatistics Unit, 34127 Trieste, Italy

Abstract. The growing popularity of Machine Learning (ML) techniques has lifted several doubts on the benefits that these approaches can offer to medical research. Machine Learning intrinsic difficulty of interpretation and lack of validation methods have limited the applicability in the clinical field. In this work, we have investigated the capability of Machine learning methods applied to survival analysis and classification in clinical context. We have compared results with the well established methods of Cox Proportional Hazards model and Logistic Regression.

Keywords: Machine Learning · Survival Analysis · Interpretability.

1 Introduction

The growing popularity of ML techniques has lifted several doubts on the benefits that these approaches can offer to medical research. ML methods have great potential to deal with complex data but their intrinsic difficulty of interpretation and lack of validation methods have limited the applicability in the clinical field in favour of classical statistics. Equally decisive are the characteristics of the available clinical datasets, often relatively limited both in the number of observations and in complexity. In this work, we provide a comparison between ML methods applied to survival analysis and classification, and the well established methods of Cox Proportional Hazards model [3] and Logistic Regression (LR). An essential aspect of the comparison is the interpretability of models, which in the clinical setting must be taken into account as much as the predictive performance. The analyses are performed on two different datasets with two aims: to develop a death risk score for High-Grade Glioma patients, and classify Sjögren's syndrome patients according to lymphoma risk.

2 Experiments

We have collected data from two multicentric studies about High-Grade Glioma [1] and Sjögren's syndrome [2]. Missing data were imputed using missForest [7, 8]. In Table 1 we present a comparison between ML methods (XGBoost [4] and Random Survival Forest (RF) [6]) with Cox Proportional Hazard model, for High-Grade Glioma dataset. ML algorithms parameters were tuned using mlr [10–15] with 10-fold-cross validation and assumptions for Cox PH model were verified. Additionally, we have used SHAP values [9] to extract both importance and direction of the impact on the risk score of the predictors.

In both classification and survival setting, ML algorithms do not outperform classical statistical techniques in terms of performance, and RF is poorly calibrated. However, ML methods provide insights about features impact on the prediction, that are comparable to statistical models and clinically plausible.

Table 1. C-index

	C-index.
XGboost	0.758
RF	0.765
Cox PH	0.762

3 Conclusions

In the analyzed context, ML algorithms do not provide substantial improvement in the prediction with respect to statistical models. However, they are able to identify important risk factors and provide useful insights on their impact on the prognosis. Future works might consider more datasets, and a deeper analysis of the interpretability of ML techniques to investigate a possible connection between the feature importance provided by ML models and Hazard Ratios as well as Odds Ratios provided by Cox PH model and Logistic Regression, respectively.

References

1. Sabatino G., Della Pepa GM., Olivi A., Pignotti F., Skrap M., Ius T.: Unpublished raw data.
2. Tzioufas A., De Vita S., Baldini C.: Unpublished raw data.
3. Cox, D.R.: Regression Models and Life-Tables. In: Journal of the Royal Statistical Society: Series B (Methodological), 34: 187-202 (1972). https://doi.org/10.1111/j.2517-6161.1972.tb00899.x
4. Chen, Tianqi, Guestrin, Carlos.: XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 785-794 (2016). https://doi.org/10.1145/2939672.2939785

5. Breiman, L.: Random Forests. In: Machine Learning 45, 5–32 (2001).
6. Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, Michael S. Lauer: Random survival forests. In: The Annals of Applied Statistics, Ann, 2(3), 841-860, (2008). https://doi.org/10.1214/08-AOAS169
7. Daniel J. Stekhoven.: missForest: Nonparametric Missing Value Imputation using Random Forest. R package version 1.4 (2013).
8. Stekhoven D. J., Buehlmann, P.: MissForest-non-parametric missing value imputation for mixed-type data. In: Bioinformatics, 28(1), 112-118 (2012).
9. Lundberg, Scott M., and Su-In Lee.: A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems (2017).
10. Bischl B., Lang M., Kotthoff L., Schiffner J., Richter J., Studerus E., Casalicchio G., Jones Z.: mlr: Machine Learning in R. In: Journal of Machine Learning Research, 17(170), 1-5 (2016).
11. Lang M., Kotthaus H., Marwedel P., Weihs C., Rahnenfuehrer J., Bischl B.: Automatic model selection for high-dimensional survival analysis. In: Journal of Statistical Computation and Simulation, 85(1), 62-76 (2016).
12. Bischl B., Kuehn T., Szepannek G.: On Class Imbalance Correction for Classification Algorithms in Credit Scoring. In: Operations Research Proceedings 2014, 37-43. Springer (2016).
13. Bischl B., Richter J., Bossek J., Horn D., Thomas J., Lang M.: mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions. In: arXiv preprint arXiv:1703.03373 (2017).
14. Probst P., Au Q., Casalicchio G., Stachl C., Bischl B.: Multilabel Classification with R Package mlr. arXiv preprint arXiv:1703.08991 (2017).
15. Casalicchio G., Bossek J., Lang M., Kirchhoff D., Kerschke P., Hofner B., Seibold H., Vanschoren J., Bischl B.: OpenML: An R package to connect to the machine learning platform OpenML. In: Computational Statistics, 1-15 (2017).

Author index

This page is intentionally left blank.