PhD-FSTM-2022-103
The Faculty of Sciences, Technology and Medecine

Dissertation

Defence held on September 07^{th}, 2022 in Luxembourg

to obtain the degree of

Docteur de l'Université du Luxembourg en Informatique

by

Salah GHAMIZI
Born on June 08^{th} 1992 in Marrakech (Morocco)

Multi-objective Robust Machine Learning For Critical Systems With Scarce Data

Dissertation defence committee

Dr. Yves Le Traon, Dissertation Supervisor
Professor, University of Luxembourg, Luxembourg, Luxembourg

Pr. Jean-philippe Thiran, Member & Reviewer
Full Professor, EPFL, Lausanne, Switzerland

Pr. Michail Papadakis, Chairman
Senior Research Scientist, University of Luxembourg, Luxembourg, Luxembourg

Dr. Maxime Cordy, Member
& Reviewer
Research Scientist, University of Luxembourg, Luxembourg, Luxembourg

Dr. Jingfeng Zhang, Member & Reviewer
Postdoctoral researcher, RIKEN AIP, Tokyo, Japan

Abstract

With the heavy reliance on Information Technologies in every aspect of our daily lives, Machine Learning (ML) models have become a cornerstone of these technologies' rapid growth and pervasiveness. In particular, the most critical and fundamental technologies that handle our economic systems are transportation, health, and even privacy. However, while these systems are becoming more effective, their complexity inherently decreases our ability to understand, test, and assess the dependability and trustworthiness of these systems. This problem becomes even more challenging under a multi-objective framework: When the ML model is required to learn multiple tasks together, behave under constrained inputs or fulfill contradicting concomitant objectives.

Our dissertation focuses on the context of robust ML under limited training data, i.e., use cases where it is costly to collect additional training data and/or label it. In the following, we study this topic under the prism of three real use cases: Fraud detection, pandemic forecasting, and chest x-ray diagnosis. Each use-case covers one of the challenges of robust ML with limited data, (1) robustness to imperceptible perturbations or (2) robustness to confounding variables. We provide a study of the challenges for each case and propose novel techniques to achieve robust learning.

As the first contribution of this dissertation, we collaborate with BGL BNP Paribas. We demonstrate that their overdraft and fraud detection systems are prima facie robust to adversarial attacks because of their feature engineering and domain constraints complexity. However, we show that gray-box attacks that consider domain knowledge can easily break their defense. We propose, **CoEva2** adversarial fine-tuning, a new defense mechanism based on multi-objective evolutionary algorithms to augment the training data and mitigate the system's vulnerabilities.

Next, we investigate how domain knowledge can protect against adversarial attacks through multi-task learning. We show that adding domain constraints in the form of additional tasks can significantly improve the robustness of models to evasion attacks, particularly for the robot navigation use case. We propose a new set of adaptive attacks and demonstrate that adversarial training combined with

such attacks can improve robustness. While the raw data available for BGL or Robot Navigation is vast, it requires a heavy cleaning, feature-engineering, and annotations by domain experts. Because this process is tedious and expensive, the end training data is scarce.

Similarly, clean labeled data is scarce when dealing with ML for medical image analysis. Therefore, our following work focuses on the challenges of robustness and generalization of Chest X-ray (CXR) classification. We first investigate the robustness and generalization of multi-task models, then demonstrate that multi-task learning, leveraging the cofounding variables, can significantly improve the generalization and robustness of CXR classification models. Our results suggest that task augmentation with additional knowledge (like extraneous variables) outperforms state-of-art data augmentation techniques in improving test and robust performances.

In contrast to the previous studies, there are cases where even raw data is scarce. It was the case when dealing with an outbreak like Covid19, and designing robust ML systems to predict, forecast, and recommend mitigation policies is challenging in the early weeks of the pandemic. In particular, for small countries like Luxembourg.

Contrary to common techniques that forecast new cases based on previous data in time series, we propose a novel surrogate-based optimization as an integrated loop. It combines a neural network prediction of the infection rate based on mobility attributes and a model-based simulation that predicts the cases and deaths. Our approach has been used by the Luxembourg government's task force and was recognized with one of the best paper awards at KDD2020.

Overall, this dissertation provides insights into the importance of domain knowledge in the robustness and generalization of models. It shows that instead of building data-hungry ML models, particularly for critical systems, a better understanding of the system as a whole and its domain constraints yields improved robustness and generalization performances. This dissertation also proposes theorems, algorithms, and frameworks to effectively assess and improve the robustness of ML systems for real-world cases and applications.

Acknowledgments

Throughout this thesis, I have received a lot of support and assistance, and I am probably forgetting many in the following.

First of all, I would like to thank Pr. Yves Le Traon who allowed me to pursue my PhD studies in his group and under his supervision. He believed in my research and provided me with valuable feedback. I am also grateful for the opportunities of teaching, working with industrial partners and with the government, and the confidence he placed in me when organizing academic activities.

Of all the people supporting my thesis, I am especially grateful to my co-supervisors Maxime Cordy and Michail Papadakis, for their patience, advice, training, and support. They taught me how to conduct proper research and present my results to others. The frequent discussions we had allowed me to better understand the field and become a better researcher.

I am also thankful to my co-authors for their efforts and feedback that contributed to making this thesis stronger. I also would like to thank the jury members for their interest in my research and the time invested in my dissertation.

I would like to express my gratitude to all my colleagues from SERVAL research group of the SnT for all the good discussions we had and also for their support during the rougher periods of my thesis. Finally, and more personally, I thank my mother and my father for their unconditional support and help throughout this enterprise. They were my never-ending source of joy and inspiration. Kudos to my friends from Mines Nancy and JCI, in particular my closest friends, Nicolas and Anas, who shared many of my burdens through this adventure.

All in all, I thank God for giving me the strength, smile, and motivation each day to enjoy my life and complete my PhD. It was a blessing and a wonderful journey.

Contents

1

Introduction

Over the last two decades, our societies have lived through profound technological changes, aided by disruptive advances on both the software and hardware fronts. The advancements in Artificial Intelligence, in particular, are happening at an exponential rate of change and are now being rolled out from academia to real industrial settings. Every day brings the announcement of a new and exciting AI application. The AI market is projected to grow to $190 billion by 2025 [FS19]. By the end of this decade, we may have progressed to Artificial General Intelligence (AGI) with its share of challenges and existential risks.

This transition did not always go smoothly and attracted the scrutiny of regulators and lawmakers. Most recently, the EU Commission has published a white paper [Com20] warning that "Artificial Intelligence (AI) entails a number of potential risks, such as opaque decision-making, gender-based or other kinds of discrimination, intrusion in our private lives or being used for criminal purposes."

I sincerely believe that while regulations have their share in solving some of these challenges, most of them can and should be addressed by the research community. I endeavored my Ph.D. to understand and contribute to solving AI's real-world challenges, precisely the challenges of Machine Learning (ML) for critical systems. The Covid-19 pandemic that struck the world during my first year of Ph.D. studies was an eye-opening and jarring experience. I was called upon to contribute to the Luxembourg Covid19 task force, which pushed my investigation toward the research scenarios where we do not have access to large data stores.

This pandemic forced us to design ML-based solutions with real-world inputs and expected actionable outputs. To this end, it required combining ML models with simulation and search techniques and investigating how to best exploit the limited data gathered from the early weeks of the pandemic in Luxembourg.

Indeed, the cases where one needs to solve complex problems with scarce real-world data are the ones where off-the-shelf machine learning models are insufficient and require considering challenges like generalization, security, and efficiency. However, they are among the cases least studied in the ML literature.

My vision with this work is to demonstrate that achieving reliable ML in critical

systems is possible if we change the paradigm of ML. No more running after even larger models or gargantuan training sets. Instead, I claim across all the chapters of this dissertation that one can achieve equivalent or superior performances by better understanding the problem and taking into account domain-knowledge that exists in the collected data or obtained with the help of experts.

In particular, this dissertation focuses on two practical challenges to ML performance that my partners have raised. The first challenge is ensuring the safety of in-production ML systems from malicious attackers. The banking institution we partnered with stated that some malicious users were actively trying to game their ML-based validation systems for fraud and money laundering. Next, we connected this practical challenge with the scientific topic of evasion attacks, where adversarial examples are generated to fool ML models. This use case encompasses challenging properties like very unbalanced classes (actual negative samples are very limited), a high cost of labeling and building training ground truths, and the expert pipeline of feature engineering and data-processing specific to this domain. Our investigation shows that off-the-shelf adversarial techniques fail, and specific designs and paradigms are required to attack and protect our partners' ML models.

The second main challenge is ensuring the generalization of ML models in medical imaging. In particular to distribution shifts and confounding factors. This work originates from a collaboration with the Hospital of Luxembourg (CHL), which was reluctant to trust SoTA models. Our evaluation demonstrates a significant drop in performance in diagnosing some pathologies when the training and testing population differ and motivated our study of a multi-task setting for this problem. Our intuition was that the techniques we previously designed to mitigate evasion attacks and improve adversarial robustness could be leveraged to improve the robustness and generalization of ML models for medical image analysis.

All in all, we present in this dissertation solutions tailored for each real-world use case but also provide an evaluation on publicly available datasets. We designed our approaches to benefit the research community as much as our partners and our contributions to transcend the use cases we considered.

For the financial sector, we show that building with the help of experts, domain-constraints within the ML system can protect the ML models from evasion attacks.

Following up on this work, we tackle the robustness of computer vision models to adversarial examples. We focus on robot navigation ML models. Inspired by the first use case, we uncover that domain-knowledge can be embedded through multi-task learning within the ML system. We demonstrate that domain-knowledge can protect the ML models from evasion attacks without additional training or data.

Moving to chest x-ray diagnosis, we outperform the SoTA in terms of robustness and generalization by leveraging auxiliary pathologies and self-supervised tasks

with limited training data. This study proposes a framework to enforce the previous findings that domain-knowledge with multi-task learning improves robustness and generalization. Furthermore, it demonstrates that we can enrich ML models with additional knowledge without new data using self-supervised learning.

Finally, we introduce an alternative form of domain-knowledge in the pandemic mitigation setting. We achieve better generalization performance with very few data points by combining time series classification, epidemiological models, and mobility meta-data.

We present this dissertation in the form of a compilation of publications. Each chapter from Chapter 4 to Chapter 8 is a published or under-review paper. While we merged and refactored the related work of all the publications in one common *"Related Work"* section (Chapter 3), we tried to preserve as much as possible the original publications' format and feel.

2

Background and Contributions

Machine learning techniques have become pervasive in our everyday life and are working their way through our most critical systems, including transportation and healthcare. While the raw performances of such approaches no longer need to be demonstrated, their robustness and generalization still pose significant challenges to practitioners. After introducing the main challenges to the robustness of ML with scarce data, we present the three use cases that our thesis covers and conclude with the contributions of our thesis.

Contents

In general, the term "robustness" is overused in the ML community, with meanings ranging from raw task performance on held-out test sets to preserving task performance on manipulated/modified inputs [DSS+21], generalization within/across domains [XM12], and resistance to malicious perturbations [YRZ+20]. While all these are important robustness goals for ML models, there are no clear definitions for robustness. [YRZ+20; DSS+21]. We focus our study on classification tasks and follow the formal definition proposed by Yanget al. [YRZ+20]:

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be an instance space equipped with a metric $\mathsf{dist} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$; this is the metric in which robustness is measured. Let $[C] = \{1, 2, \ldots, C\}$ denote the set of possible labels with $C \geq 2$. For a function $f : \mathcal{X} \to \mathbb{R}^C$, let $f(\mathbf{x})_i$ denote the value of the ith coordinate.

Definition 1. ***Robustness and Astuteness:*** *Let $\mathbb{B}(\mathbf{x}, \epsilon)$ denote a ball of radius $\epsilon > 0$ around $\mathbf{x}$ in a metric space. We use $\mathbb{B}_\infty$ to denote the ℓ_∞ ball. A classifier g is* robust *at $\mathbf{x}$ with radius $\epsilon > 0$ if for all $\mathbf{x}' \in \mathbb{B}(\mathbf{x}, \epsilon)$, we have $g(\mathbf{x}') = g(\mathbf{x})$. Also, g is* astute *at $(\mathbf{x}, y)$ if $g(\mathbf{x}') = y$ for all $\mathbf{x}' \in \mathbb{B}(\mathbf{x}, \epsilon)$. The* astuteness *of g at radius $\epsilon > 0$ under a distribution μ is*

$$\Pr_{(\mathbf{x},y)\sim\mu}[g(\mathbf{x}') = y \text{ for all } \mathbf{x}' \in \mathbb{B}(\mathbf{x}, \epsilon)].$$

The goal of robust classification is to find a g with the highest astuteness. We sometimes use *clean accuracy* to refer to standard test accuracy (no perturbation), in order to differentiate it from *robust accuracy* a.k.a. astuteness (with perturbation or distribution shifts).

2.1 Challenges of Robust Machine learning

We first introduce the main challenges to robust learning through the prism of causal reasoning. We summarize these challenges in Fig. 2.1. *Data scarcity* (Fig. 2.1 a) refers to the lack of high-quality training data needed to build effective models. *Data mismatch* occurs when a model developed in a controlled environment fails to generalize to real-world data. It happens either because the test population P_{te} has shifted from the training population P_{tr} (Fig. 2.1 b), or the prevalence of each label has changed (Fig. 2.1 c). emphData mismatch also occurs when the selected samples mismatch the overall distribution (Fig. 2.1 d), for example, during data sampling.

Castro et al. [CWG20] proposed a causal perspective on the robustness of ML models to tackle these challenges. In practice, it assumes we can represent the data and label generation process as a Structural Causal Model (SCM): a Directed Acyclic Graph (DAG) where the nodes are the variables, and the SCM represents causal relationships among these variables using structural equations.

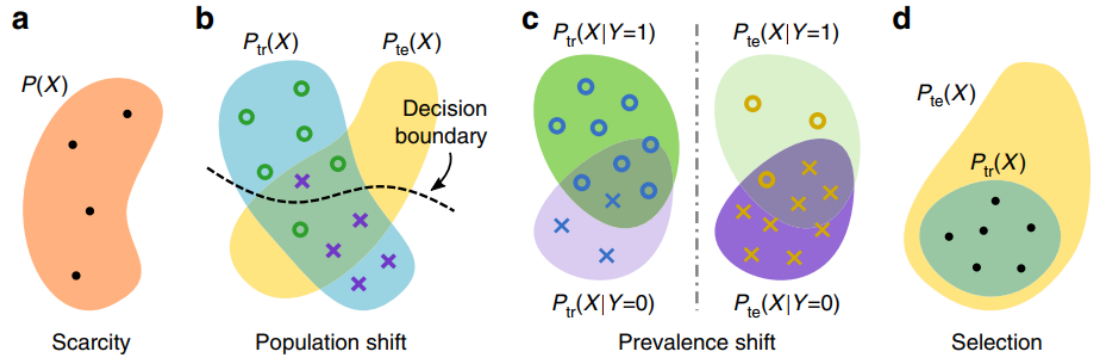


Figure 2.1: Challenges to Robust ML according to Castrol et al. [CWG20]. P, P_{tr}, and P_{te} refer to the total population, the training population and the test population.

These equations link each node to each parent and to an exogenous noise term (denoted by ϵ).

We show, for instance, in Fig. 2.2 a Structural Causal Model for the chest x-ray medical imaging case. In particular, in orange, we highlight the **population shifts** characterized by poorly explored factors like confounding variables. In blue, we showcase the **acquisition shift**, where the collection protocol (including imperceptible changes) can affect the variables. In green is the **sample selection** where the actual training data can arbitrarily be selected and in red is the **annotation shift**, where the data is labeled automatically or with domain experts. Within this SCM, the ML task can be *causal* like predicting the annotations in the image such as segmentation labels Y4 (red arrows in Fig. 2.2). The ML task can be *anti-causal* in the case of predicting the diagnosis of the patient Y1 (blue arrows in Fig. 2.2).

In this thesis, we study the acquisition shift through the lens of adversarial perturbation. I.e., imperceptible perturbation to the images designed to cause misclassification. Next, we investigate the annotation shift with surrogate models for building the ground truths of our DL models. Finally, we mitigate the **population shifts** by collecting the confounding attributes and augmenting our models with multi-task learning.

2.1.1 Robustness to Imperceptible Perturbations

The phenomena of *evasion attacks* has first been introduced by Biggio et al. [BCM+13] and Szegedy et al. [SZS+13]. It refers to the process of generating imperceptible perturbations in the input to cause a misclassification. It has since gathered the interest of researchers to propose new attacks [GSS14a; MMS+17b], defense mechanisms [KGB16; HWC+17], detection mechanisms [MGF+17], or to improve transferability across different networks [TPG+17; IWL+19].

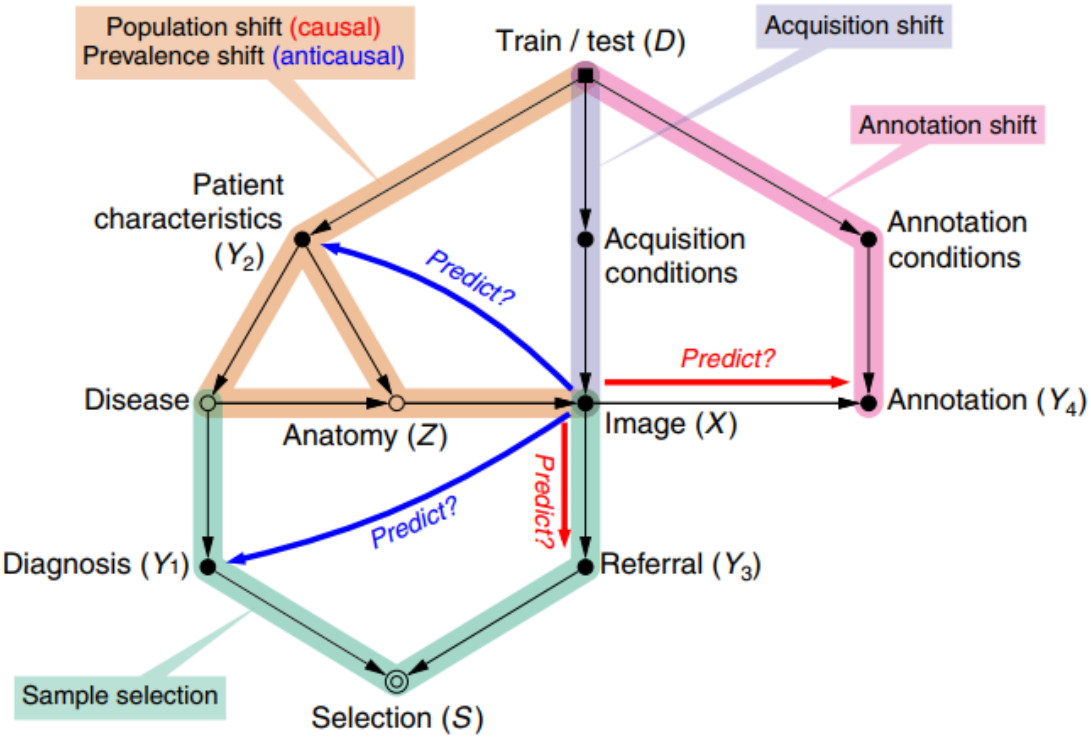


Figure 2.2: Structural Causal Model for chest x-ray medical imaging predictions proposed by Castro et al. [CWG20]. Starting from the Image (X), we can either perform anticausal predictions (blue arrows) to predict pathologies (Y1) or to predict patient characteristics such as gender or age (Y2). In both cases, the predicted attribute is the cause of the features present in the input features X. We can also perform causal predictions (red arrows), for example, to decide to refer the case to an expert (Y3) or segment areas of the image (Y4). In both cases, the output is directly dependent of the pixels in the radiograph X.

Initially focused on image classification, research on "evasion attacks" has since demonstrated that the adversarial threat affects a wide range of domains, including cybersecurity [PPC+20; SPW+20], natural language processing (NLP) [ASE+18], cyber-physical systems (CPS) [LLY+20], finance [GCG+20b], manufacturing [MH20b], robot navigation [GCP+21c] and others.

However, each domain requires a different definition of *imperceptible perturbations*. In NLP, the changes can affect letters or words. However, the perturbed sentences should preserve the semantic meaning of the original example a human oracle gave and remain realistic inputs, i.e., grammatically correct. For finance, the generated examples should respect the domain constraints such as the client's

balance, typology, and previously aggregated financial data related to the attacked transaction.

Consequently, researchers have recently focused on domain-specific evasion attacks that either perturb the ML features in ways that satisfy the domain requirements. We call these *constrained feature-space attacks*. Others went one step further and designed attacks that affect real objects (as opposed to ML features) via a series of problem-space transformations. We refer to these attacks in the following as *problem-space attacks*. These attacks provide examples that are realistic by design but are computationally more expensive to run compared to standard attacks.

We evaluate the robustness of ML systems to imperceptible perturbations with unconstrained feature-space attacks for the medical imaging use-case and the robot navigation use-case. We also evaluate the robustness of constrained feature-space attacks in financial use-case and provide techniques for each scenario to significantly improve ML systems' robustness.

2.1.2 Robustness to Confounding Variables

A *confounder* is a variable that affects the relationship between input data (e.g., images) and output variables (e.g., classification labels), causing the results to differ from the accurate prediction.

Improper modeling of those relationships often results in spurious and biased associations. There are several methods for excluding or controlling confounding factors, including randomization, restriction, and matching [PBV12]. In randomization, the random sampling of study subjects is assumed to break any links between the data and confounders by generating groups of samples that are reasonably comparable in terms of known and unknown confounding variables. In restriction, one eliminates the variation in the confounder by selecting samples with the same confounding values. Finally, matching approaches try associating each confounding factor with a pair of control. For instance, if gender is the confounding factor, ensure that the sample of one gender has a matching sample of the opposite gender.

However, all those approaches require collecting large amounts of data to sample/restrict/match while keeping sufficient training data for each output variable. Acquiring such an amount of data is hardly feasible in ML under scarce data settings, for instance, for medical imaging or early-stage pandemic forecasting.

We propose to mitigate the impact of confounding factors in two manners: In the case of ML for medical imaging, we propose to include the confounding factors as additional tasks of the model. We demonstrate that this approach outperforms existing approaches for clean and robust performances. In the case of the Covid19 pandemic, we propose to extend the model's features with surrogate variables. We use mobility data as proxies for the intensity of interactions of the population and demographic data as a proxy for the stratification of the population. We

demonstrate that extending the time-series learning with these variables mitigate the biases associated with confounder and improves the robustness of the learning as a whole.

2.2 Machine Learning Under Scarce Data Setting

We introduce in the following the real-world scenarios where we evaluated the robustness of ML systems with limited data, and proposed novel algorithms and approaches to improve their performance.

2.2.1 ML for Financial Services

The banking industry increasingly relies on machine learning to support decision-making based on customers' historical data. One prominent application case is *credit scoring*, i.e., "a set of decision models and their underlying techniques that aid credit lenders in the granting of credit" [TEC02]. By learning from history – credit cases and their outcomes (whether the credit was returned in time) – supervised models can automate the approval and rejection of new credit requests with limited human intervention.

Our industrial partner, the Data Science Lab of BGL BNP Paribas Luxembourg (henceforth referred to as "BGL BNP Paribas"), has recently engineered such a credit scoring system. Their system deals with approving overdraft requests, which occur when a transaction causes the account balance to drop below zero. Then, it is up to the bank employees to allow or reject this transaction. BGL BNP Paribas implemented an automated system relying on random forests. The approval of overdraft requests is seen as a binary classification problem (approved or rejected). The system approves or rejects overdrafts automatically, based on data about the requested transaction and customer history. If the system rejects the overdraft, an expert re-analyzes the request and may overrule the decision. If it is accepted, the system later checks whether the overdraft has been reimbursed in time.

The first challenges faced by our partner were feature engineering and model selection. As these have been widely researched (see, e.g., [dMB16; SCF18; FLH+19]), they benefited from the available body of knowledge and techniques to build a quality system that achieved a test accuracy of 80%. As of now, this system has processed more than 400,000 overdraft requests over 30 months. However, these transactions cover a large pool of clients with varying typologies and behaviors. The currently collected and labeled data is far from sufficient to cover the behaviors of all the clients and their evolution through time.

Yet, the stringent security requirements forced upon the banking sector oblige them to protect their credit scoring system against malicious third parties. In our partner's context, the threat lies in the capability of the third party to modify the requested loans and the profile of customers to make the system accept overdrafts

that it should have rejected.

Typically, the process of changing the features into malicious inputs is called an *evasion attack*. The adversarial inputs are crafted by altering benign inputs in such a way that they fool the classification system. Adversarial examples are mainly studied in computer vision, and deep neural networks [SZS+13; BR18; AM18], where elusive pixel alterations of images cause misclassifications.

Contrary to the rich literature of adversarial examples in DL for computer vision, the application of evasion attacks to FinTech and random forests remains largely unexplored [PMJ+16]. This scarce research literature is surprising given the widespread use of these techniques in industrial applications. Evasion attacks generally lean on the internal computations of the classification models. Therefore, they disregard the fact that altering the original input may produce false positives, i.e., *infeasible* examples in the real world or *invalid* examples given the acceptable inputs for the ML-based system. While this phenomenon is less likely to occur in image recognition, where slightly altering an image can quickly produce a valid image, application domains such as FinTech are subject to hard domain constraints delimiting the set of valid inputs. For instance, a credit scoring system relies on financial information such as customers' account balance, contracted credits, monthly income, and indebtment rate. Such data are naturally constrained (e.g., income is positive), interdependent (indebtment rate depends on contracted credits and monthly income), or bounded (e.g., the maximum overdraft amount authorized by the bank). Thus, any successful attack should respect these domain constraints and produce examples that satisfy them.

Supporting domain constraints is a recurrent problem in software engineering [AIA+13]. In the case of generating adversarial examples, one cannot handle/satisfy the domain constraints independently of the attack technique. The issue is that on top of the constraints (many of which are imposed by other systems/components), one needs to craft the attacks and fulfill some additional objectives (e.g., cause misclassification, maximize the overdraft amount). Therefore, reducing the problem to constraint satisfaction is not satisfactory.

To deal with this issue, we propose a search-based method that generates constrained adversarial examples for banking applications. We formulate the generation of adversarial examples satisfying the domain constraints as a multi-objective search problem and show that search-based techniques offer suitable solutions. Our method, called **Constrained Evolutionary Adversarial Attack (CoEvA2)**, operates in a grey-box way; it relies on the feature representation of the inputs but is independent of the internal parameters of the classification model.

We apply CoEvA2 to BGL BNP Paribas's credit scoring system. We show that it can generate thirteen thousand valid adversarial examples and that up to 8.45% of the real overdrafts transaction can be corrupted to cause a misclassification.

Our approach drastically improves over state-of-the-art evasion attacks, which failed utterly. Next, we show that we can make our partner's system more robust by performing adversarial training (i.e., retrain the model using the produced adversarial examples). After such training, the system resists our attack (applied under similar conditions) and remains robust to existing attacks.

2.2.2 ML for Medical Image Diagnosis

In the clinical setting, artificial intelligence (AI) decision support systems (DSS) are designed to assist radiologists in maintaining diagnostic performance in the face of growing clinical loads. Medical errors, especially diagnostic errors, are accounted for additional medical spending of $17 to $29 billions [KCD+00]. More recent studies have shown that misdiagnosis errors of chest x-ray images remain high even with the advances in practice and imaging systems [BCG18].

The challenge of providing a reliable and efficient diagnosis has motivated increasing research for automated diagnosis systems. While the first attempt for an automated CXR diagnosis system started in the 1960s [LKD63], recent techniques using Machine Learning (ML) have shown promising performance [RIZ+17; YPC+19]. Today, the U.S. Food and Drug Administration has cleared about 200 AI medical products related to radiology and other imaging domains [1]. While these systems provide remarkable figures in their respective studies, recent research has shown discrepancies in their actual performances [YPC+19; PBB20; BNG+19]. Other studies put forward a few hypotheses to explain the discrepancies: Errors in labeling [Oak19], practitioners' biases and disagreements [BCG18] and, more generally, overfitting of models and lack of generalization across multiple datasets [CHB+20].

Indeed, deploying an ML model in a target population different from its training population may result in exploiting data for training and evaluation with different distributions, thus, breaking the independent and identically assumption [DH20]. An ML model is said to be *generalizable* from a source to a target population when its performance metrics on the target population do not significantly drop compared to its performance on the train and test populations. Generalizability is essential in the medical domain because practitioners need models that provide stable predictions and can efficiently adapt to new clinical settings (e.g., different hospitals, different populations, etc.) at an affordable computation cost. Lack of generalization hampers the safe and accurate translation into clinical trials ([BBa20]).

Experimentally, we can assess model robustness through the prism of imperceptible perturbation. The study of the adversarial vulnerability of image classification models has only recently tackled medical systems. However, the few studies of

[1] `https://aicentral.acrdsi.org/`

chest x-ray (CXR) classification robustness [FKB18; TDH18; MNG+21; LZ20] have focused on binary classification (healthy VS diseased) and drew conclusions from one dataset and one or two models. We argue that these previous techniques to assess and improve CXR models' robustness are inconsistent and flawed. We propose a new framework to improve adversarial robustness, **Adversarial Training with Task Augmentation (ATTA)**, that leverages knowledge from concurring pathologies and confounding factors to enhance the adversarial and generalization robustness of models.

2.2.3 ML for Pandemic Forecasting

As depicted in Section 2.2.2, achieving high ML performance on the local data without assessing the generalization to slightly different datasets is useless in practice. It is particularly the case for time-series forecasting, where the models are trained on past data and expected to behave effectively on unseen future data. Pandemic outbreaks prediction is a typical time-series forecasting task that suffers from acute training data scarcity.

At the early stage of the outbreak of the COVID-19 pandemic, the world has been facing a human tragedy with overwhelmed healthcare systems and fears of economic collapse. In the absence of vaccines to immunize the population rapidly at scale, governments have implemented various non-pharmaceutical public health interventions such as social distancing and lockdowns. Considering that the World Health Organisation (WHO) was foreseeing the first clinical trials of a vaccine for the end of the year 2020 [Wor20], decision-makers had to plan their exit strategies carefully. Afterward, the measures to contain the spread were to be methodically lifted to avoid the risk of precipitating new outbreaks.

In this context, mathematical modeling offers public health planners frameworks to make predictions about the spread of emerging diseases and assess the impact of possible mitigation strategies.

There are two main kinds of models: *static cohort models* and *transmission dynamic models* [JB11]. Static models typically rely on decision trees and Markov processes and assume a strength of infection independent of the proportion of the infected population. Transmission dynamic models, on the other hand, see the force of infection vary depending on the proportion of the infected population. Compared with static cohort models, transmission dynamic models are usually more complex to parameterize and require epidemiological information on the infectious disease and demographic and economic information about the affected population.

Different techniques exist to implement dynamic approaches. Agent-Based Models (ABM) are simulations composed of agents that interact with each other and their environment. Because each agent can follow its own rules, this approach can capture aggregate phenomena derived from the behavior of single agents. These models offer a great explainability of the root causes leading to the propagation of a

disease but are computationally intensive to run. Thus, they are hardly applicable to large populations. Indeed, the behavior and the interaction of each type of agent need to be fully defined in order for the model to be helpful. These rules are case-specific and are not transferable from one population to another.

The alternative technique to implement dynamic approaches is to rely on epidemiological models. The most common approach to model the spread of infectious disease is the Susceptible-Infected-Removed (SIR) model and its extension *SEIR* (Susceptible, Exposed, Infectious, and Recovered). These are state-based models, and every state expresses the degree of exposure of a population to the disease. These models are equation-based, where each equation defines the rate to go from one state to the other. The SEIR model thus separates the population into four groups and simulates the evolution over time of each subpopulation. The transition rates are defined by (1) the time scale to which an individual can transmit the disease, (2) the time to recovery, and (3) the number of newly infected people due to an infected individual. The most varying parameter is the *effective reproduction number (R_t)*. It expresses the number of people an infectious individual can contaminate over time.

These methods depend on the hyper-parameters validity, like the reliability of the transition rates. While SEIR is a compelling model, it presents a significant limitation; it requires hyper-parameters that are hard to observe, such as the infection rate of an individual. In practice, SEIR parameters are manually set to fit with the local observations to the considered population (e.g., country) and are not learned from larger-scale observations. To circumvent the limitation of such epidemiological models, researchers started to take advantage of the advances made in Machine Learning (ML) to create models based on available large datasets [VMU+20b; SCC+20a]. We name this family of approaches **ML-based epidemiological models**.

As part of the government's task force to provide DSS to the policymakers, we focused on building a forecasting model for the number of cases and deaths in Luxembourg, even when we only had a few hundred training samples. To achieve high performance, we extend the models' inputs by leveraging the studied populations' demographic and economic data. These confounding variables affect both the original inputs of our model (i.e., mobility behaviors of the population) and the output of our model (i.e., infection rate). We also rely on surrogate models for the output: a SEIR model to simulate the casualties from the predicted infection rate. This approach, **DN-SEIR**, alleviates manual tuning of the SEIR model and achieves remarkable generalization performances across different countries and time splits.

Next, we combined our model with an evolutionary search loop to explore and optimize various exit strategies and constraints. We evaluate three common

hand-crafted exit strategies and show that multi-objective genetic algorithms combined with DN-SEIR can find atypical Pareto-front strategies that minimize death numbers and economic impact.

2.3 Contributions

We formally present the contributions of this dissertation as follows:

1. **A novel framework for adversarial robustness assessment and defense of a constrained ML system in production (Chapter 4).** We assess the robustness of an overdraft authorization system at BGL BNP Paribas. We aim to investigate if the currently deployed system is robust against adversarial attacks under a realistic threat model and, if negative, how to improve its robust performances. We show that existing attacks are inapplicable to real-world credit scoring systems like our partner's. In doing so, we demonstrate the need for domain-constrained adversarial attack techniques for industrial and financial systems. To this end, we develop **CoEvA2**, a new adversarial attack method (for random forest applications) based on a multi-objective search. Given a classification model and domain constraints, CoEvA2 effectively generates valid adversarial examples. We evaluate CoEvA2 on our partner's system and empirically show that it can craft adversarial examples with an actual success rate of 8.45%, leading to thousands of examples.

2. **A large-scale empirical study of adversarial robustness of multi-task models (Chapter 5).** We investigate the factors that may explain the robustness of multi-task models to adversarial attacks. We refine the theory of robust multi-task learning through the concept of *marginal adversarial vulnerability* of tasks. Leaning on this, we demonstrate that the inherent vulnerability of tasks plays a central role in the model's robustness. Following this theoretical analysis, we empirically show that a careful weighting of the tasks can act as a remedy and offer the benefits initially promised by previous research. However, it does not provide increased robustness against adaptive attacks. However, determining which combination of tasks is optimal can be costly given a target model. We propose different surrogates to approximate the gain in robustness and show that they strongly correlate with the robustness of the target model.

3. **A novel and efficient improvement to Adversarial Training (Chapter 6).** Based on the intuition that domain constraints can improve the robustness of ML models, as demonstrated in our first study, we build on top of our second study a novel approach to robustify any deep learning

model using auxiliary tasks. Our approach, **Adversarial Training with Task Augmentation (ATTA)**, adds carefully-chosen auxiliary tasks to an original single-task model during the min-max optimization of AT. We evaluate our approach with limited-training data and demonstrate that on CIFAR-10 classification, robust accuracy improves up to four times (from 11% to 42%). On the CheXpert medical image dataset, ATTA improves the robustness of scarce pathologies from 50% to 83%. Next, we evaluate ATTA using typical image classification datasets and demonstrate that on the full CIFAR-10 dataset, ATTA with adaptive weighting outperforms all data augmentation strategies. Furthermore, we can combine ATTA with different data augmentation strategies to enhance robust accuracy from 22% to 48%.

4. **An extensive study of how Task Augmentation can also improve generalization of Chest X-ray classification models (Chapter 7).** While Task Augmentation can be used with adversarial training to improve the robustness to adversarial attacks, we hypothesize that it can also improve the robustness of models to confounding factors and hence, improve the generalization to unseen datasets. We demonstrate that learning multiple pathologies in a multi-task model can significantly increase or decrease generalization performance (-10% to +10% AUC-ROC) on each pathology. To measure this, we train ML classifiers on all pairs of pathology among seven (using the source dataset) and assess their performance on the target dataset. Our experiments reveal that some pathologies consistently improve generalization regardless of the pathology they are jointly learned with. Based on these results, we propose **Auxiliary Pathology Learning**, a method to improve medical model generalization via a multi-task model that simultaneously learns the main pathology of interest with a well-chosen auxiliary pathology. Next, we extend our approach to support the gender and age confounding factors and demonstrate that when we leverage these factors in our framework, the generalization performance can significantly be improved. Finally, we show that Auxiliary Pathology Learning achieves competitive generalization across various pathologies using only 6% to 34% training data of the SoTA techniques.

5. **A robust and generalizable ML framework for pandemic forecasting using surrogate models and confounding attributes (Chapter 8).** At the heart of the COVID-19 crisis, governments have taken drastic solutions to stop the pandemic, including restrictions and lockdown measures, which put the economy at a standstill. In order to support the decision-makers in their forecasting of the pandemic evolution and devising mitigation strategies, we develop in tight collaboration with the local authorities a pipeline and

a set of tools. First, we design **DN-SEIR**, a novel approach that alleviates manual tuning of epidemiological models by relying on confounding attributes and surrogate models. Our approach combines the *SEIR* epidemiological model with a machine learning regression model to estimate the *effective reproduction number (R_t)* over time. The machine learning predictor relies on demography and mobility features to improve the generalization of the predictor. For each time increment, R_t is updated and used for the next timestamp of the epidemiological model. We evaluate our approach for twelve countries worldwide and show that our approach that mixes demographics, mobility, and epidemiological data provides better forecasts for 9 out of 12 studied countries than existing epidemiological models. Next, we extend this prediction model into a simulation tool for policymakers. The simulator enables them to design mitigation plans, like lockdown schedules and activity restrictions, then assess their impact in terms of hospitalization, infected people, deaths, GDP growth, and employment per economic sector. Our simulator can also automatically explore and optimize various mitigation plans, objectives, and constraints. We evaluate three hand-crafted mitigation plans deployed by different countries and show that multi-objective genetic algorithms can find atypical Pareto-front strategies that minimize death numbers and economic impact.

3

Related Work

In this chapter, we first present the literature related to evasion attacks *and adversarial robustness. Subsequently, we address the robustness of ML systems to distribution shifts. Next, we focus on the prior work proposed to achieve effective multi-objective learning, particularly multi-task learning approaches. Finally, we conclude this literature review with the prior work related to the robustness challenges to ML systems for the three use-cases of our manuscript: ML for finance, ML for chest x-ray classification, and ML for pandemic forecasting. During this overview, we address the shortcomings of the existing work and position our contributions accordingly.*

Contents

3.1 Evasion attacks and defenses

An *adversarial attack* is the process of intentionally introducing perturbations on the inputs of a machine learning model to cause wrong predictions. One family of adversarial attacks is *poisoning attacks* [BNL12] where the inputs targeted are the training set and occur during the learning step, while *evasion attacks* [BCM+13] focus on the inference step.

One of the earliest attacks is the Fast Gradient Sign Method (FGSM) [GSS14a]. It adds a small perturbation η to the input of a neural network, which is defined as:

$$\eta = \epsilon \operatorname{sign}(\nabla_x \mathscr{L}_i(\theta, x, y_i)), \tag{3.1}$$

where θ are the parameters of the network, x is the input data, y_i is its associated target, $\mathscr{L}(\theta, x, y_i)$ is the loss function used, and ϵ the strength of the attack.

Following Goodfellow, other attacks were proposed, first by adding iterations (I-FGSM) [KGB16], projections and random restart (PGD) [MMS+17b], momentum (MIM) [DLP+18] and constraints (CoEva2) [GCG+20a; DGS+22].

These algorithms can be used without any change on a multi-task model if the attacker only focuses on a single task.

Adversarial training (AT) AT is a method for learning networks that are robust to adversarial attacks. Given a multi-task model $\mathscr{M}_\theta$ parameterized by θ, a dataset $\{(x_i, y_i)\}$, a loss function $\mathscr{L}$ and a perturbation space Δ, the learning problem is cast as the following optimization problem:

$$\min_\theta \sum_i \max_{\delta \in \Delta} \mathscr{L}(\theta, x_i + \delta, y_i). \tag{3.2}$$

A typical choice for a perturbation space is to take $\Delta = \{\delta : \|\delta\|_\infty \leq \epsilon\}$ for some $\epsilon > 0$. This is the ℓ_∞ threat model used by [MMS+17c] and is the setting we study in this paper. The procedure for AT is to use some adversarial attack to approximate the inner maximization over Δ, followed by some variation of gradient descent on the model parameters θ.

AT has since been improved, for example, to balance the trade-off between standard and robust performances like TRADES [ZYJ+19] and FAT [ZXH+20], or to speed up the training [SNG+19; WRK20]. Finally, AT has been combined with data augmentation techniques, either with unlabeled data [CRS+19], self-supervised pre-training [CLC+20], Mixup data augmentation [RGC+21] or data from generative networks [GRW+21].

These last approaches showed significant improvements over the literature but entail a computation overhead that can be prohibitive for common cases and even impossible when training data (even unlabeled) is scarce or when generative networks are non-conclusive like medical imaging applications.

3.2 Distribution shifts

In domain shift literature, the challenge is to learn a machine learning model that can generalize to unseen data distributions and test environments. It has been thoroughly studied outside the medical context, in particular for computer vision tasks [WLL+21].

The naive baseline is to use empirical risk minimization (ERM) to learn on a mixture of data across all training environments. Recent approaches involve augmenting the training data, for instance using adversarial data augmentations [ZYH+20], or optimizing the learning strategies: Ensemble Learning [WG21], Meta-Learning [KLP+21], and Self-supervised Learning [CDB+19]. Meanwhile, Representation Learning has proven to be fruitful in increasing generalization performances. It encompasses techniques that aim for domain-invariant representation learning [HZC+19; JLZ+20; MMW+20] and techniques that investigate feature disentanglement [PLS20].

3.3 Multi-task learning

Multi-task learning leverages shared knowledge across multiple tasks to learn models with higher efficiency [VGV+21; SZC+20]. A multi-task model is commonly made of an encoder block that learns shared parameters across the tasks and a decoder part that branches out into task-specific heads.

Vandenhende et al. [VGV+21] recently proposed a new taxonomy of MTL approaches to split the approaches based on where the task interactions occur. They differentiated between approaches that are *encoder-focused* where some information is shared across tasks at the encoder stage and approaches that are *decoder-focused* where some interactions still happen across the heads of the tasks. They organized MTL research around three main questions: (1) when does the task learning interact, (2) how can we optimize the learning, and (3) which tasks should be learned together.

To answer the first question, MTL networks historically belonged either to the "Hard Parameter sharing" family, where we restrict the parameter sharing to the encoder, and each head of the decoder freely learns its own parameters. Some of the approaches of this family are UberNet [Kok16], Stochastic Filter Grouping [BTO+19]. The second family is the "Soft Parameter sharing" family, where the heads still have some level of interactions and constraints, and the tasks interact from the early stages of the networks. Popular approaches of this family include Cross-stitch Networks [MSG+16] that propose linear combinations of the activations similar to skip connections of ResNet architectures. Contrary to Cross-stitch Networks, NDDR-CNN [GMZ+19] suggests to heuristically share features on some specific layers to leverage common knowledge. Recently, research

investigated attention mechanisms for MTL. Approaches like MTAN [LJD19] use soft-attention modules for each task and exhibit state-of-the-art performances while being less sensitive to various weighting schemes.

To answer (2), Vandenhende surveyed the different optimization techniques from the literature and pointed out that most optimizations focus on task balancing. In the MTL context, we need to consider that the loss of individual tasks may be imbalanced. Large unbalances may cause some tasks to dominate the joint learning process and harm the learning of the other tasks. Some of the most common strategies are to choose ad-hoc weights based on fine-tuning, use Gradient normalization [CBL+18], Uncertainty weighing [KGC18] or Multi-objective optimizations [SK18].

The weighting strategy to choose remains an open challenge and depends on the studied task dataset. Gong et al. [GLS+19] and Leang et al. [LSB+20] showed that there was no clear winner and similar performance among strategies, including a uniform weighting strategy.

While (1) and (2) have extensive literature with well-established approaches, answering (3) remains challenging. Prior work evaluated the similarity of tasks based on the transfer-learning performance from one to another [**taskonomy**; DR19; WTW19]. More recent work showed that learning transfer affinity and multi-task learning affinity display notable differences[SZC+20].

Our work is the first to tackle task selection through the prism of robustness, not learning affinity.

Adversarial attacks on multi-task models. The problem formulation of adversarial attacks for multi-task models involves taking into account the summed loss across all the tasks. Given a multi-task model $\mathscr{M}_\theta$ parameterized by θ, an input example x, and its corresponding ground-truth label $\bar{y}$, the attacker seeks the perturbation δ that will maximize the joint loss function of the attacked tasks – i.e., the summed loss, within a p-norm bounded distance ϵ. The objective of the attack is then:

$$\underset{\delta}{\operatorname{argmax}} \mathscr{L}(\theta, x + \delta, \bar{y}) \text{ s.t. } \|\delta\| p \leq \epsilon \tag{3.3}$$

Mao et al. [MGN+20] were the first to evaluate these gradient-based attacks. They extended the first-order vulnerability found by [SOB+19] and showed that increasing the number of tasks leads to more robust models.

Our work builds on this to propose a novel framework to extend any existing single-task model into a multi-task model to improve both the robustness to shifts and adversarial examples.

3.4 ML for finance

In this thesis, we address the case of automating credit and overdraft approval, which falls under the *credit scoring* family of problems. Credit scoring is defined as "a set of decision models and their underlying techniques that aid credit lenders in the granting of credit" [TEC02]. While credit scoring models are widely researched, the robustness of the obtained models remains largely unexplored.

Credit Scoring The study of Louzada et al. [LAF16] presents a comprehensive survey of classification methods in the context of credit scoring automation. While focusing on classification models, Louzada et al. also reported the different problems tackled by the surveyed papers, none related to model robustness or adversarial attacks.

Research on feature engineering and model selection for credit scoring is abundant. For example, De Melo and Banzhaf [dMB16] combined Kaizen programming and logistic regression to find the best non-linear combination of features. Saia et al. [SCF18] proposed a wavelet-based feature engineering method and evaluated its performance using multiple models. Feng et al. [FLH+19] proposed a feature selection approach based on filters and a novel index named *new separation degree*.

However, multiple questions naturally follow once the features are selected, and the model trained. How stable is the model over time? Can the changes in the economic situation or people's habits render the model unusable? Is it possible to trick the system and make a client look trustworthy when it is not the case? Or is it possible to attack the model in another way - i.e., reduce the performance of the whole model by introducing some carefully engineered credit histories into the dataset? These and similar questions relate to the general concept of *model robustness*, and our study of credit-scoring models mainly focuses on their robustness to evasion attacks.

Evasion attacks on Financial ML systems Papernot et al. [PMJ+16] mentions the potential threats of evasion attacks and adversarial examples for financial fraud detection. Yet, to the best of our knowledge, there existed no prior work applying evasion attacks to industrial systems from the financial domain at the time of the study.

In a follow-up research [PMG16], Papernot et al. present an attack against decision trees. While this attack can easily extend to random forests commonly used for credit scoring tasks, it does not support domain constraints. As we show in Chapter 4, this attack cannot generate adversarial examples satisfying the domain constraints.

Kantchelian et al. [KTJ16] have also proposed another random forest attack. It transforms the decision nodes into binary and algebraic formulas and uses a *SAT solver* to generate solutions following the misclassification objective. Thus, any

solution corresponds to an adversarial example. While this attack can theoretically solve the problem of generating constrained adversarial examples (by adding constraints into the formulas), it faces scalability issues in practice due to the expensive nature and limitations of SAT solvers.

Indeed, we conducted an exploratory experiment based on the HELOC dataset[1] which has half the number of features compared to our partner's dataset and simple constraints involving at most two features. After 20 hours, the Kantchelian attack could not generate any adversarial example satisfying the constraints.

Our method overcomes the scalability limitations using search-based (evolutionary) algorithms to generate adversarial examples that satisfy the domain constraints. The idea of using search-based algorithms to perform adversarial attacks is not new. Alzantot et al.[ASC+18] have proposed a black-box attack on image recognition models (viz., deep neural networks). Being focused on images, the problem they tackle is different and does not involve domain constraints.

Constrained Input Generation The problem of generating inputs under domain constraints is not new [MS96] and was tackled by several works in the context of traditional (code-based) software, as witnessed by the survey of McMinn [McM04]. More recently, Ali et al. citeAliIAB13 evaluated different search-based methods in generating test inputs satisfying the constraints. In the context of Combinatorial Interaction Testing (CIT), Garvin et al. [GCD09] proposed reorganizing the search space of metaheuristics to reflect the structure of the CIT problems and their inherent constraints. Compared to such works, the novelty of our research is that it targets machine learning systems in an adversarial setting.

3.5 ML for medical diagnosis

Chest x-ray (CXR) is an affordable, easy-to-use medical imaging and diagnostic technique. It is commonly used to diagnose a broad range of lung diseases and abnormalities, such as Atelectasis, Pneumothorax, and even early lung cancer. Chest film reading consists of identifying areas of increased density or areas of decreased density. Identifying each area can be tasked to ML systems for classification or image segmentation tasks. While the literature about ML for CXR classification is abundant, we focus below on the work related to the two challenges covered by this dissertation: Adversarial robustness and domain generalization.

Evasion attacks in medical imaging. Taghanaki et al. [TDH18] were among the first to evaluate the robustness of CXR image classification against adversarial examples. They evaluated both white box and black box attacks on two binary classification neural networks (ResnetV2 and NasNet Large) on the ChestX-ray14 dataset [WPL+17a]. They showed that both models are vulnerable to gradient-

[1] `https://community.fico.com/s/explainable-machine-learning-challenge`

based attacks (100% success rate of attacks). While their evaluation pioneered the research on adversarial attacks in the medical setting, their evaluation focused on binary classification, which is rarely the focus of CXR medical imaging classifiers.

Finlayson et al. [FKB18] focused on binary image classification for medical diagnosis. Their study covered CXR, Fundoscopy, and Dermoscopy diagnosis, and they also showed that PGD attacks achieved 100% success rate on the ChestX-ray14 Pneumothorax label using one model. While their study is comprehensive, it evaluates unrealistic settings and perturbation budgets.

Ma et al. [MNG+21] had another take on the robustness of CXR image classification models. They compared the robustness of binary classification, 3-label, and 4-label classification. They showed that while PGD had a success rate of over 99% on all of them, the vulnerability seems to decrease with the increased number of labels. Our study covers the entire scenario of 14-labels classifiers trained on different datasets (i.e., population distributions) and architectures. A previous study [CHB+20] showed significant performance differences across the different labels that hint that an overall evaluation of the robustness of the model can be misleading. Indeed, some labels are already challenging to learn, and we hypothesize they will be similarly easier to attack than others.

Contrary to the abovementioned work, we show that CXR models can be resilient against adversarial attacks when the right pathologies are learned together.

Domain generalization in medical imaging. Domain shift and its impacts on generalization performance are even more acute when dealing with the medical imaging context.

A model trained on hospitals in one region may be deployed to another. However, due to domain shift (*e.g.*, differences in the age of patients), prediction performances may drop, leading to erroneous diagnoses. Tackling Chest X-ray classification, Zhang et al. [Za21] investigate how subsampled datasets with varying label prevalence between genders can impact the generalization capabilities of the models and their impact on fairness and bias. Mahajan et al. [MTS21] show that the class-conditional domain invariant objective for representations is insufficient. Specifically, when the distribution of the stable features to be learned varies across domains, the class-conditional objective is insufficient to learn the stable features.

Similarly, Pooch et al. [PBB20] evaluate how well models trained on a hospital-specific dataset generalize to unseen data from other hospitals. Their work uncovers the drop in generalization performance of the models. However, it does not provide any insight into the causes or how to mitigate this drop (except by selecting one dataset over another). Cohen et al. [CHB+20] study the cross-domain performance and agreement between models. They identify discrepancies between the performance and agreement of models. Then, they provide insights on the representation changes from one dataset to another. While their evaluation is exhaustive, it pro-

motes the common assumption that generalization is attained by mixing different datasets through training or ensemble. Our work is parallel to theirs. We do not study the latent representation change from *one dataset* to another, but the latent representation change from a *combination of pathologies* to another.

To the best of our knowledge, our work is the first to investigate the impact of multiple pathology learning and its implications on robust and generalization performances.

3.6 ML for pandemic forecasting

Machine Learning approaches have been widely used to model and forecast former epidemics, especially to handle the increasing complexity of the epidemiological models underneath. Popular approaches remain regression trees and forests [MSI+18], and neural networks [KM18]. While there is a plethora of reports that tackle COVID-19 forecasting, the peer-reviewed literature about ML and COVID-19 is rather scarce. Most approaches in public repositories tackle ML regression in combination with the SIR epidemiological model [AMG+20] or its SEIR extension [PCG+20; YZW+20; SCC+20b]. However, their models depend on the hyper-parameters and the quality of the past data used for regression. Indeed, the large differences in the pandemic data obtained from different countries make existing approaches ineffective in tackling a peculiar country like Luxembourg, with its size, data scarcity, and residents' mobility patterns.

In parallel to time series forecasting, recent research proposed using mobility data to enrich models and learning algorithms. In [VMU+20a], Vollmer et al. integrate mobility in a stochastic model. They focus on Italy and suggest that COVID-19 transmission rate and mobility metrics are closely related.

Our approach relies on a similar intuition, taken one step further. To overcome the distribution shift across the COVID-19 data of different countries and train one model using a mixture of all the data, we enrich our forecasting model with confounding features related to the demographics and socio-economic attributes of the countries used for training. In addition to these country-specific attributes, we incorporate additional features related to mobility data and cartography, and demonstrate that incorporating this additional domain-knowledge significantly improves the generalization performance of our model. Especially for countries with scarce COVID-19 data at that time. Countries like Luxembourg.

4

Search-Based Adversarial Testing and Improvement of Constrained Credit Scoring Systems.

Credit scoring systems are critical FinTech applications that concern the analysis of the creditworthiness of a person or organization. While decisions were previously based on human expertise, they are now increasingly relying on data analysis and machine learning. In this chapter, we assess the ability of state-of-the-art adversarial machine learning to craft attacks on a real-world credit scoring system: We study the industrial case of the ML models deployed by our industrial partner, BGL BNP Paribas. We find that, while these techniques can generate large numbers of adversarial data, these are practically useless as they all violate domain-specific constraints. In other words, the generated examples are all false positives as they cannot occur in practice. To circumvent this limitation, we propose CoEvA2, a search-based method that generates valid adversarial examples satisfying the domain constraints. CoEvA2 utilizes multi-objective search in order to simultaneously handle constraints, perform the attack and maximize the overdraft amount requested. On BGL's production system, CoEvA2 generates thousands of valid adversarial examples, revealing a high risk for the banking system. We provide all our models, and open source-code at `https://github.com/serval-uni-lu/coeva2`.

Contents

4.1 Introduction

The banking industry increasingly relies on machine learning to support decision making based on customers' historical data. One prominent application case is *credit scoring*, i.e., "a set of decision models and their underlying techniques that aid credit lenders in the granting of credit" [TEC02]. By learning from history – credit cases and their outcomes (whether the credit was returned in time) – supervised models can automate the approval and rejection of new credit requests with limited human intervention.

Our industrial partner, the Data Science Lab of BGL BNP Paribas Luxembourg (henceforth referred to as "BGL BNP Paribas") has recently engineered such a credit scoring system. Their system deals with the approval of overdraft requests, which occur when a transaction causes the balance of the account to drop below zero. Then, it is up to the bank employees to allow or reject this transaction. BGL BNP Paribas implemented an automated system relying on random forests. That is, the approval of overdraft requests is seen as a binary classification problem (approved or rejected). The system approves or rejects overdrafts automatically, based on data about the requested transaction and the customers' history. If the overdraft is rejected by the system, an expert re-analyzes the request and may overrule the decision. If it is accepted, the system later checks whether the overdraft has been reimbursed in time.

The first challenges faced by our partner were feature engineering and model selection. As these have been widely researched (see, e.g., [dMB16; SCF18; FLH+19]), they benefited from the available body of knowledge and techniques to build a quality system that achieved a test accuracy of 80%. As of now, this system has processed more than 400,000 overdraft requests over a span of 30 months.

Yet, the stringent security requirements forced upon the banking sector oblige them to protect their credit scoring system against malicious third parties. In our partner's context, the threat lies in the capability of the third-party to modify the requested credits and the profile of customers to make the system accept overdrafts that it should have rejected.

In machine learning, such malicious inputs are called *adversarial examples* and are crafted by altering benign inputs in such a way that they fool the classification system. Adversarial examples are mainly studied in the context of computer vision and deep neural networks [SZS+13; BR18; AM18], where elusive alterations to the pixels of images cause misclassifications. Such research has shown that adversarial examples can be crafted by a systematic procedure – the *adversarial attack* – which typically utilizes information about the neural network's gradients to find the slightest perturbation that would change the output class.

Interestingly, the application of adversarial attacks to FinTech and random forests remains largely unexplored [PMJ+16]. This is surprising given the widespread

use of these techniques in industrial applications. To our knowledge, the state-of-the-art attack for random forest classification algorithms is the one designed by Papernot et al. [PMG16]. It consists of a stochastic procedure that visits and attempts to flip the individual decision nodes of the forest's trees until the classification outcome is changed. An alternative approach could be to built a "surrogate" deep neural network (using the training data), based on which we could apply a prominent gradient-based attack (with the hope that this attack will be transferable to the random forest model).

Nevertheless, all adversarial attack techniques lean on the internal computations of the classification models and disregard the fact that altering the original input may produce false positives, i.e., *infeasible* in the real world, or *invalid* for the software system inputs that are acceptable by the classification model. While this phenomenon is less likely to occur in image recognition, where slightly altering an image can easily produce a valid image, application domains such as FinTech are subject to hard domain constraints delimiting the set of valid inputs. For instance, a credit scoring system relies on financial information such as customers' account balance, contracted credits, monthly income, and indebtment rate. Such data are naturally constrained (e.g., income is positive), interdependent (indebtment rate depends on contracted credits and monthly income) or bounded (e.g., the maximum overdraft amount authorized by the bank). Thus, any successful attack should respect these domain constraints and produce examples that satisfy them.

Moreover, we conduct experiments with the current state-of-the-art, i.e., the Papernot attack, on our partner's system.[1] Interestingly, we show that while the attack successfully generated adversarial examples that flipped the classification results for 75% of the cases (its *gross* success rate), none of them satisfied the domain constraints. This means that the attack has an *actual* success rate of 0%. These results indicate that state-of-the-art adversarial attacks cannot generate domain-constrained test inputs.

Dealing with domain constraints is a recurrent problem in software engineering [AIA+13]. In the case of generating adversarial examples, one cannot handle/satisfy the domain constraints independently of the attack technique. The issue is that on top of the constraints (many of which are imposed by other systems/components), one needs to craft the attacks and fulfil some additional objectives (e.g. cause misclassification, maximize the overdraft amount). Therefore, reducing the problem to constraint satisfaction is not enough.

To deal with this issue, we propose a search-based method that generates constrained adversarial examples for banking applications. We formulate the generation of adversarial examples satisfying the domain constraints as a multi-objective search problem and show that search-based techniques offer suitable solutions. Our

[1] We report on these experiments in Section 4.2.4.

method, called Constrained Evolutionary Adversarial Attack (CoEvA2), operates in a grey-box way; it relies on the feature representation of the inputs but is independent of the internal parameters of the classification model.

We apply CoEvA2 to BGL BNP Paribas's credit scoring system and show that it can generate thirteen thousand of valid adversarial examples from 8.45% of the real overdrafts. This drastically improves over state-of-the-art adversarial attacks, which failed completely. Then, we show that we can make our partner's system more robust by performing adversarial training (i.e., retrain the model using the produced adversarial examples). After such training, the system resists to our attack (applied under similar conditions).

In summary, the contributions of this study are:

- We demonstrate the need for domain-constrained adversarial attack techniques for industrial financial systems. We also show that existing attacks are inapplicable to real-world credit scoring systems, such as the one of our partner.
- We develop CoEvA2, a new adversarial attack method (for random forest applications) based on multi-objective search. Given a classification model and domain constraints, CoEvA2 effectively generates valid adversarial examples.
- We evaluate CoEvA2 on our partner's system and empirically show that it can craft adversarial examples with an actual success rate of 8.45%, leading to thousands of examples.
- We demonstrate that our method helps to improve the system's robustness (to adversarial attacks). Indeed, retraining the system on adversarial examples results in improving its robustness significantly.

4.1.1 Industrial Credit Scoring System

Process and Datasets

When a customer initiates, through any channel, a transaction whose amount exceeds the customer's account balance, the payment engine asks the credit scoring system (CSS) for permission. The CSS examines the customer's profile and either approves the credit overdraft or it suggests the operator reject the request. In the latter case, the operator can follow the suggestion of the CSS or overrule it and accept the request.

To make informed decisions, the CSS pulls information from a dozen sources. In addition to basic features like the transaction amount and the customer's current balance, much information about the customer's history is consolidated. In the end, an overdraft request is represented as a vector of 46 features.

After approving an overdraft request, the bank expects the customer to return the credit in due time. In case the customer does not do so, the bank considers that it was wrong to allow the overdraft; otherwise it considers that it was correct.

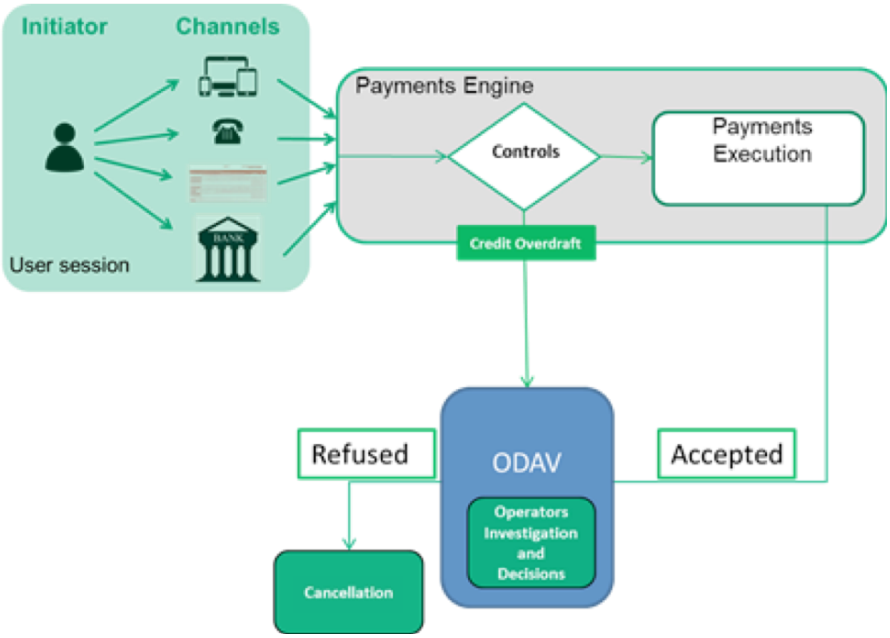


Figure 4.1: Overview of the overdraft approval process

Through this post-analysis, we can associate each approved overdraft credit with a binary label (true or false). Such labels form the *ground truth* and are used to assess the accuracy of the CSS. A similar process is used for rejected overdraft and analyzes, based on the customer's future transactions, whether the overdraft credit would have been returned in time should it have been approved.

Overall, the CSS dataset comprises 400,000 overdraft credit requests with their associated label, out of which 275,000 are used for training and 125,000 for testing. We show in Figure 4.1 the overdraft approaval process.

Model Requirements and Characteristics

The rationale behind our partner's project is to reduce human intervention in overdraft approval by automatically approving safe overdraft requests (sending only rejected overdraft to human experts) while minimizing the acceptance of risky overdrafts (e.g. transactions of large amount). Our partner also expects the system to run online, in real-time, and efficiently so that it does not compromise the efficiency of the other services. Finally, the selected model should be *interpretable*, as explaining hardly-interpretable models can be inefficient and even dangerous in high-stake decision-making processes [Rud19] such as overdraft approval.

To satisfy those requirements, our partner performed feature engineering in close collaboration with business experts. They performed model selection (considering decision trees, random forest and gradient boosted trees) and used grid search to find optimal model parameters. AUC for ROC curve was used as an optimization criterion for the grid search, while F1 score was the criterion to choose the optimal

classification threshold. The final model is a random forest with 500 estimators up to 8-level deep.

This model is built and integrated within a Dataiku DSS pipeline [2]. It achieves acceptable performance: 0.99 AUC and 0.99 accuracy on the training set; 0.88 AUC, 0.80 accuracy and 0.70 F1-score on the test set.

4.2 Problem Formulation

4.2.1 Unconstrained Adversarial Attack

Let $f(.)$ be a binary classification model defined over a input space I. For simplicity, assume I to be normalized such that $I = [0..1]^m$ and $f(i) \in \{1, 0\}$ for any $i \in I$. Let $\mathbf{x_0} \in I$ represents an original example correctly classified by $f(.)$.

Adversarial attacks generate altered inputs that are close to their original counterparts, yet are misclassified by the model. In traditional, unconstrained adversarial attacks, the ideal adversarial example $\mathbf{x}^*$ crafted from $\mathbf{x_0}$ to fool $f(.)$ is defined as:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{x_0}\|_p$$

such that

$$f(\mathbf{x}) = 1 - f(\mathbf{x_0})$$
$$\{\mathbf{x}, \mathbf{x_0}\} \subset I$$

and where $\|.\|_p$ is the L_p norm (e.g. L_2).

The p-norm distance between a perturbed input and an initial one is a good first indication of the effort required to generate the adversarial example. However, to be acceptable, the perturbed input has to satisfy inherent domain constraints. This is a fundamental difference with image recognition, where it is generally admitted that a small distance between $\mathbf{x}$ and $\mathbf{x_0}$ ensures that $\mathbf{x}$ has a strong perceptual similarity to $\mathbf{x_0}$ and, thus, constitutes a valid image.

Therefore, the problem of generating adversarial attacks for ML-based FinTech systems takes a different form: both the original and the adversarial examples must be part of the subspace of inputs that are considered valid. To characterize this subspace, we proceed by first eliciting the different domain constraints.

4.2.2 Formalization of the Constraints

A first validity criterion demands that the adversarial example still represents an overdraft, that is, the transaction amount remains above the current balance of the customer's account. Additionally, we consider this amount relevant if it is

[2] https://www.dataiku.com/

higher than 1,000.00 currency units. Features[3] can also be interdependent. For instance, the indebtment rate must be positive and is obtained by dividing the monthly credit reimbursement by the monthly income. There also exist categorical features that can only take values from a finite set. For example, each customer can be associated with a personal level of risk (e.g. on a 1–10 scale) based on its profile and past interviews with the bank. The corresponding feature can only take as value any integer between 1 and 10.

This highlights the types of constraints that our method must support: features can be bounded, each by a different bound, some may only take certain values, and there may exist numerical dependencies between them. Accordingly, we define that a formula ϕ encoding such constraints (i.e. a constraint formula) over a set F of features is formed according to the following grammar:

$$\phi := \phi_1 \wedge \phi_2 \mid\mid f \succeq \psi \mid f \in \{c_1 \dots c_k\}$$
$$\psi := c \mid f \mid \psi_1 \oplus \psi_2$$

where $f \in F$; $c, c_1, \dots, c_k$ are constant values; ϕ, ϕ_1, ϕ_2 are constraint formulae; $\succeq \in \{<, \leq, =, \neq, \geq, >\}$; ψ, ψ_1, ψ_2 are numerical formulae and $\oplus \in \{+, -, *, /\}$.

In addition to satisfying such formula, an adversarial attack may not be able to modify some features. For instance, the level of risk associated to a customer is under control of the bank and cannot be changed by the customer himself. The same holds for features resulting from the aggregation of data over time. Thus, we enforce the requirement that the attack can only alter the subset $\mathcal{F} \subseteq F$ of *mutable* features. The other features (which the attack cannot alter) are *immutable*. In BGL BNP Paribas's CSS, 16 features are mutable and the other 30 are immutable. This means that the attacker's capability to succeed strongly depends on the 30 features it cannot change. We thus consider the features of different customers as different starting points for our search algorithm.

We cannot disclose the features of our partner's model, or its specific constraints, but we provide in the GIT repository a replication example on the Lending Club Load Datase[4]

We can however describe the various types of constraints that we extracted with our partner from the engineered features of their pipelines: Let x represent the feature vector, and x_i its ith feature.

Transaction-based constraints Within our dataset, many features are computed directly based on the history of the transaction of an account. The most

[3]Due to NDA we cannot reveal the exact features used. The examples of feature we provide are different from the ones used by our partner. However, their interrelations are of the same level of complexity.

[4]https://www.kaggle.com/wendykan/lending-club-loan-data

Table 4.1: List of features of our classifier

Feature set	Notation	# Features
Transactions	$\mathcal{T}$	40
Online	$\mathcal{O}$	7
Cat/Numerical	$\mathcal{C}$	9
Hierarchical	$\mathcal{H}$	3
Attacked	$\mathcal{A} = (\mathcal{T} \cap \mathcal{O}) \cup \mathcal{C}$	16
Total		46

obvious ones are the balance of each of the clients account (current, savings, ...) and the trends over multiples days / weeks. We denote the set of indices of the *raw* transaction-based features by $\mathcal{T}$.

Online-based constraints The feature preparation process that builds the vector x runs over different temporalities. Some features are computed as a batch every day while others are updated at each new transaction. We define the set of indices of features that are updated in real time as $\mathcal{O}$.

Categorical and numerical constraints Contrary to adversarial examples in image classification, where all the features are real numbers, our industrial case contains features that identify clients and accounts, or can only take a very limited set of input values. Some features are therefore one-hot encoded. We group all the indices of those features in the set $\mathcal{C}$.

Hierarchical constraints Finally, some constraints describe the transactions at different level. We can use the client identifier as a level, we can also use his individual accounts as a lower-level, or a cluster identifier as a higher-level. Every level contains a set of possible lower-level values.

In table 4.1 we present these features where we restrict our attacks to 16 features (set $\mathcal{A}$), only taking into account the features updated in real-time after every transaction and the features constrained by the new transaction.

4.2.3 Constrained Adversarial Attack

Let $I = [0..1]^m$ be the feature vector space over the feature set $F = \{f_1 \dots f_m\}$, $\mathcal{F} \subseteq F$ be the set of mutable features and ϕ be the formula over F encoding the domain constraints. Furthermore, let I_ϕ denote the subspace of valid feature vector, i.e. $I_\phi = \{i \in I : i \models \phi\}$.

Given a binary classification model $f(.)$ and an original input $\mathbf{x_0} = \{(\mathbf{x_0})_1, \dots (\mathbf{x_0})_m\} \in I_\phi$, the ideal adversarial example $\mathbf{x}^*$ generated from $\mathbf{x_0}$ to fool $f(.)$ is defined as

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{x_0}\|_p$$

such that

$$
\begin{aligned}
f(\mathbf{x}) &= 1 - f(\mathbf{x_0}) \\
\{\mathbf{x}, \mathbf{x_0}\} &\subset I_\phi \\
f_i \notin \mathcal{F} &\Rightarrow (\mathbf{x_0})_i = (\mathbf{x})_i, \forall 1 \leq i \leq m.
\end{aligned}
$$

Here, the difficulty of performing such attack lies in that it can only alter features in $\mathcal{F}$ and in a way that ϕ remains satisfied.

4.2.4 Motivation: How helpful are existing attack techniques?

We start our study by assessing the capability of existing (unconstrained) attacks to generate valid adversarial examples in our real-world use case. We assess the gross success rate of these attacks (percentage of times they manage to create an example misclassified by the model), their actual success rate (after removing the examples that do not satisfy the domain constraints) and the average amount of perturbation applied (measured as the L_2 distance to the original input). The amount of perturbation is meant to serve as a metric comparison between the attacks.

4.2.5 Random Forest Attack

First, we consider the attack proposed by Papernot et al. [PMG16] – henceforth named the *Papernot attack* – which was originally designed to cause misclassifications in decision trees by visiting all nodes in the tree and making them flip until the misclassification is achieved.

This is the only attack relevant to our case. So, we adapt it (to random forests) by iteratively applying the Papernot attack to every tree of the forest until the classification outcome of the random forest changes. We call this method *Iterative Papernot*.

We evaluated *Iterative Papernot* on all original test inputs of our use case where the model makes correct classifications. The results are recorded in Table 4.2 and reveal that the attack seems successful, as it manages to generate adversarial examples (causing misclassification) in 74.86% of our starting points/inputs, with an average L_2 distance (to the corresponding original inputs) of 10.64. However, it turned out that none of the generated inputs satisfied the domain constraints, leading to an actual success rate of 0%.

4.2.6 Gradient-Based Attacks

Another popular family of adversarial attacks are the gradient-based attacks. These attacks were designed to generate adversarial examples on Deep Neural Networks (DNNs). We note that during learning, a DNN iteratively adjusts its

Table 4.2: Success rates and average perturbation produced by existing adversarial attacks applied on our partner's system. While every method manages to generate adversarial examples, none of these satisfy the domain constrains.

Attack	Gross success rate	Actual success rate	Avg L_2
Papernot	74.86%	**0.00%**	10.64
PGD	17.30%	**0.00%**	0.10
CW2	80.00%	**0.00%**	0.37

neurons' weight according to the gradient of its cost function (which depends on the weights). Gradient-based attacks exploit the same information to produce a perturbation that changes the output of the last neuron layer, thereby changing the classification outcome.

Being gradient-based, those methods can apply only on models relying on differentiable cost functions. Thus, they do not work *out of the box* on random forests.

A common way to circumvent this limitation is to build a *surrogate* model (a DNN) that mimics the random forest. That is, we train this DNN on the same input set and use the outputs (classification results) of the random forest as the ground truth for the DNN. Then, we perform the gradient-based attack on the surrogate DNN and obtain an adversarial example. The underlying assumption of this method is that any adversarial example that fools the DNN also fools the mimicked model.

For our experiments, we consider two gradient-based attacks: Projected Gradient Descent (PGD) [MMS+17a] and CW2 [CW17], which are considered among the most effective attacks. We apply each attack on all original test inputs that the model correctly classifies. We implement a DNN model using the Tensorflow/Keras frameworks and we use the implementation of the gradient-based attacks provided by the IBM robustness library [NST+18].

Results are shown in Table 4.2. PGD succeeds in generating adversarial examples (causing misclassification) in only 17.30% of the attempts, yet it does so with the smallest average amount of perturbation amongst all techniques (L_2 distance of 0.10). Nevertheless, none of the generated adversarial satisfy the domain constraints. CW2 has a much higher gross success rate (80%) at the cost of a higher perturbation than PGD (0.37), yet much lower than the Papernot attack. Like the other two methods, CW2 fails to generate a single example satisfying the constraints.

Overall, our analysis shows that, by focusing on classification method and outcome, while being unaware of the domain constraints, **state-of-the-art attacks**

fail to generate valid adversarial examples. This fact demonstrates the need for new constraint-aware attacks, i.e., attacks that satisfy the constraints *by design*.

4.2.7 Research Questions

Having shown that state-of-the-art adversarial attacks are not useful in our case, we look for ways to circumvent their limitations and successfully generate valid adversarial examples. We focus more particularly on the use case where a malicious third party aims at fooling the system, i.e., making it approve overdrafts that should be rejected. This use case is deemed relevant by our partner as it induces a risk of financial loss for the bank.

To this end, we investigate whether simple methods satisfying the domain constraints can solve our problem. Thus, in our first question, we check whether altering the initial points while keeping constraints satisfied is sufficient. Hence, we ask:

RQ1 *Can we generate successful adversarial examples by just satisfying the domain constraints?*

To answer this question, we investigate two solutions. The first is to extend the Iterative Papernot attack in order to make it consider the constraints as it searches through the nodes. The second is to search for solutions (using single objective search) that satisfy the constraints. Then, we can check whether the produced examples are adversarial.

As our results shall show, these single-objective methods can craft examples that either change the classification outcome or satisfy the domain constraints, but not both at the same time. We conjecture that, on the one hand, the iterative nature of the Papernot attack blocks it into a narrow part of the landscape and, on the other hand, the random search does not benefit from the knowledge of the original input (causing arbitrary perturbation). This means that an effective search should not only be guided with additional criteria (e.g., minimize the perturbation) but also explore a diverse space.

To achieve this, we experimented with evolutionary (genetic) algorithms. Such techniques are directed by some feedback, aka fitness function, that quantifies how close the current solutions are to the sought ones. At the same time, the random alterations they apply to the candidate solutions create disruption in the search and, doing so, avoids falling into local optima.

Our definition of constrained adversarial attack (see Section 4.2.3) hints that such a search algorithm needs to handle multiple objectives: minimize perturbation, flip the classification, satisfy the domain constraints (changing only mutable features). Additionally, a malicious third party looks for optimizing a *domain-specific objective*: maximize the overdraft amount.

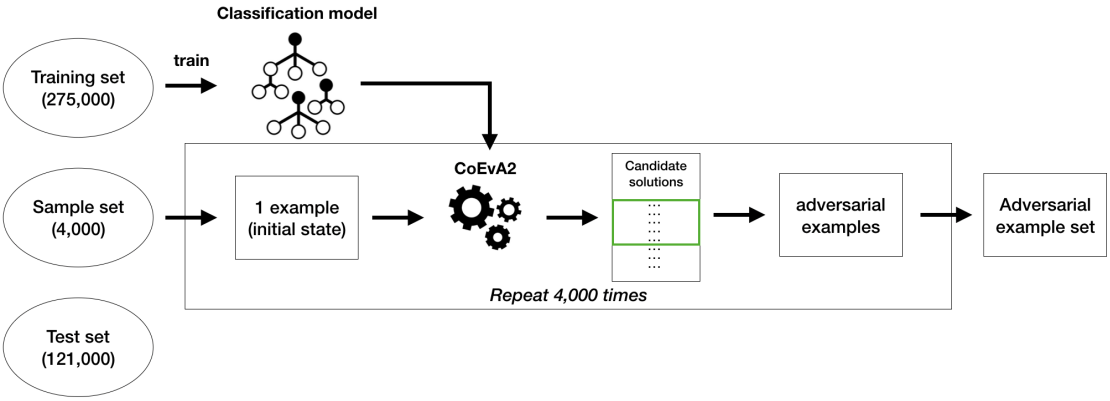


Figure 4.2: Overview of CoEvA2. Adversarial examples are generated from benign inputs (sampled from the test set).

Thus, we design a genetic algorithm that handles all these constraints and objectives. We assess its performance and investigate, in particular, which fitness function (combination of objectives) performs the best. Thus, we ask:

RQ2 *How effective is our fitness function at generating constrained adversarial examples?*

We answer this question by presenting our algorithm, named *CoEvA2*, and empirically evaluating it using different variants of the fitness function.

Having shown that our method constitutes an effective attack, we aim to improve the defence mechanism of our partner's system, in order to eliminate any risk that real-world attacks succeed. Therefore, we turn our attention toward improving the robustness of the CSS. To achieve this, we used *adversarial training*, which consists of re-training the model with generated (successful) adversarial examples, together with their correct classification label. Such a practice is widely popular and has been shown to improve the robustness of machine learning models. We, therefore, use adversarial training to improve our partner's system and check the scale of this improvement. Hence, we ask:

RQ3 *How much adversarial training based on CoEvA2 can increase the robustness of the system?*

We answer this question by checking the success rate of CoEvA2 when applied on various starting points.

4.3 Search-based Generation of Constrained Adversarial Examples

Figure 4.2 displays an overview of the CoEvA2 process. Starting from a set of samples (randomly selected from the test set), we iterate over the elements of this set. At each iteration, CoEvA2 starts from the sampled element (named the *initial state*) and creates an initial population of new examples. Then, it evolves this population with the aim of finding valid adversarial examples.

4.3.1 Population

Since only a subset of the features are mutable (16 out of 46 in our industrial case), an adversarial example can differ from the initial state only by the value of its mutable features. Thus, given an initial state s and a feature vector space I, the **population** is a subset $P \subset I$ of feature vectors such that any **individual** $p \in P$ has the same value as s for all immutable features. We can, therefore, reduce the **genotype** of an individual as a single **chromosome**, which is the vector of its mutable features. Any **gene** is an element of this chromosome and contains the value of the corresponding mutable features.

Note that we do not require any individual to satisfy the domain constraints ϕ or to cause a misclassification. Indeed, we allow the algorithm to produce invalid and benign examples throughout the evolution process. This provides a smooth landscape for the search, allowing it to explore efficiently this large search space. Constraint satisfaction and misclassification are actually encoded into the fitness/objective functions (see Section 4.3.2), in a way that valid adversarial examples are considered better than invalid and benign ones. Since misclassification is one of the objective, the evaluation of the individuals makes use of the attacked model (in a black-box way, using only the output class probabilities).

4.3.2 Fitness Function

We formulate the generation of constrained adversarial examples as an optimization problem with four objectives. Each objective can be independently assessed through an objective function.

The first objective function f_1 models the requirements of causing misclassification, that is, maximizing the probability that the example is classified in the targeted class. It is defined as the distance between the example and the incorrect class targeted by the adversarial attack.

Without loss of generality we assume the target class is 0 (the correct class is 1). When provided with an input $\mathbf{x}$, a binary classification model outputs $p(\mathbf{x})$, the prediction probability that $\mathbf{x}$ lies in class 1. If $p(\mathbf{x})$ is above the classification threshold (a hyperparameter of the model), the model classifies it in class 1;

otherwise, in class 0. Thus, we see $p(\mathbf{x})$ as the distance of $\mathbf{x}$ to class 0. By seeking an input $\mathbf{x}^*$ that minimizes this distance, we increase the likelihood of misclassification *regardless* of the actual classification threshold. Thus, we have:

$$f_1(\mathbf{x}) = p(\mathbf{x}).$$

The second objective is to minimize the amount of perturbation measured between the initial state and the adversarial example, a common requirement of adversarial attacks [BCM+13]. We use a conventional measure of this amount: the normalized L_2 distance between the two inputs. Thus, given an initial state $\mathbf{x_0}$, the distance from an example $\mathbf{x}$ and $\mathbf{x_0}$ is given by

$$f_2(\mathbf{x}) = L_2(\mathbf{x}, \mathbf{x0}).$$

The third objective is the actual domain objective, that is, maximizing the approved overdraft credit amount. By convenience, we transform this objective into a (normalized) minimization problem. Let $\mathbf{x}$ be an example and $(\mathbf{x})_t$ be the value of the feature encoding the requested overdraft amount. Then, the objective function f_3 can be defined as

$$f_3(\mathbf{x}) = \frac{1}{(\mathbf{x})_t}$$

Thus, this objective considers that the most successful solution is the one that reaches the highest overdraft amount. In practice, though, our partner (like most banks) specifies a maximal overdraft amount above which the transaction is always rejected.

The fourth and last objective concerns the satisfaction of the domain constraints. As mentioned, we allow individuals to violate the constraints as the evolution progresses. Yet, to converge towards valid adversarial examples, we transform the satisfaction of each (numerical) constraint into a (normalized) *penalty* function to minimize, representing how far an example $\mathbf{x}$ is from satisfying the constraint. More precisely, we transform each constraint into an inequality of the form of $C(X) \geq 0$ (e.g. $3f \geq g$ yields $3f - g \geq 0$). If the constraint is not satisfied, $C(X) < 0$ and we use the absolute value of C(X) as distance. The overall distance to constraint satisfaction is the mean of the normalized individual distances.

Thus, assuming $\phi = \bigwedge_{i=1..k} \phi_i$, the fourth objective function is defined as:

$$f_4(\mathbf{x}) = \frac{1}{k} \sum_{\phi_i} penalty(\mathbf{x}, \phi_i).$$

In our implementation, the transformation of the constraints into these penalty functions is automatically handled by the framework we use (see more in Section 4.3.4). Other heuristics to compute such distance to satisfaction exist [MS96; McM04] and could be considered in future work.

Overall, we consider that the success of an adversarial example can be measured by the trade-off between the likelihood of flipping the classification outcome, the applied perturbation, the overdraft amount and the satisfaction of the constraints. To objectively quantify this trade-off, we define our fitness function as a linear equation over the four objective functions, that is:

$$fitness(\mathbf{x}) = \alpha \times f_1(\mathbf{x}) + \beta \times f_2(\mathbf{x}) + \gamma \times f_3(\mathbf{x}) + \delta \times f_4(\mathbf{x})$$

where $\alpha, \beta, \gamma, \delta > 0$ are meta-parameters that specify the relative importance of the four objective. Overall the search process will attempt to generate examples that minimize this fitness function and simultaneously fulfil the four objectives.

In practice, we set these meta-parameters according to our partner's requirements and experience. The rationale was to reflect the domain requirements:

- Constraints: These shape the valid input space, meaning that any non-conforming input is invalid/infeasible. It is imperative to satisfy the constraints and hence, we make them our most important objective ($\delta = 1,000$).
- Maximise overdraft: For a bank, minimising the potential loss of money is of utmost importance. Indeed, the overdraft amount represents the potential gain for the attacker, which forms the objective to maximize ($\gamma = 100$).
- Cause misclassification: we also deemed it more important to cause misclassification than to minimizing perturbation ($\alpha = 2$, $\beta = 1$), such that the perturbation should only serve to rank adversarial examples that are successful and valid.

As revealed by our experiments, our fitness function provides a feasible and practical solution to our problem. Alternatively, we could have relied on search methods to automatically set the weights. Another option is to define four fitness functions (one per objective), thereby reducing our problem to multi-objective optimization and search for Pareto fronts. While studying these alternatives is of interest, it is unlikely that they will make major differences under such interdependent constraints. The github repository proposes both a grid-search optimisation of the weights and a non-dominated multi-objective approach (NSGA-2) and shows limited performance improvements in comparison with the weights proposed by our domain-expert.

4.3.3 Generation Process

Algorithm 1 formalizes the generation process of our genetic algorithm. From a given initial state $\mathbf{x_0}$, we generate an initial population P including L individuals, by randomly setting the mutable features of $\mathbf{x_0}$ (Line 1). The only constraints we enforce are the categorical constraints of the form $f \in \{c,1 \ldots, c_k\}$ and the boundary constraints of the form $f \succeq c$ where f is a feature, $\succeq \in \{<, \leq, =, \neq, \geq, >\}$, and $c, c_1, \ldots, c_k$ are constant values. This allows reducing the number of invalid

Input: $\mathbf{x_0}$, an initial state;
$fitness$, a fitness function;
N_{gen}, a number of generations;
L, a population size;
Output: A population P of adversarial examples minimizing the $fitness$ function;

1 $P \leftarrow init(\mathbf{x_0}, L)$;
2 **for** $j = 1$ *to* N_{gen} **do**
3 $\quad fit \leftarrow \emptyset$;
4 $\quad X \leftarrow x_0$;
5 $\quad X[mutableMask] \leftarrow P$;
6 $\quad fitness\_prediction \leftarrow f1(X, x_0)$;
7 $\quad fitness\_perturbation \leftarrow f2(X, x_0)$;
8 $\quad fitness\_overdraft \leftarrow f3(X)$;
9 $\quad loss\_constraints \leftarrow f4(X)$;
10 $\quad fitness \leftarrow fitness\_prediction + fitness\_perturbation + fitness\_overdraf + loss\_constraints$;
11 $\quad P_{survive} \leftarrow binary\_tournament\_select(P, fitness)$;
12 $\quad P_{offspring} \leftarrow SBX\_crossover(P_{survive})$;
13 $\quad P \leftarrow P_{survive} \cup polyMutate(P_{offspring})$;
14 **end**
15 **return** P

Algorithm 1: Generation process of CoEvA2

examples without biasing the generation (since the boundary constraints involve only one feature each).

Then, we make the population evolve for a predefined number N_{gen} of generations (Lines 2–13). At each iteration (generation), we evaluate the fitness function of each individual of the current population P. This is achieved by, first, combining the genotype of each individual (its mutable features) with the immutable features of $\mathbf{x_0}$. Then, we can input any resulting example $\mathbf{x}$ into the *fitness* function (as defined previously) and obtain the fitness value of $\mathbf{x}$.

What follows is the application of selectors and alterers to form the next generation. We first use tournament selection that keeps the best individuals (according to the fitness function) out of samples of two (Line 11). Thus, half of the population disappear.

As for alterers, we randomly apply crossover and mutation operators. For the crossover (Line 12), we randomly pick pairs of individuals (that survived the tournament selection) and use a simulated binary crossover [DSO07] to create two new offsprings from the numerical and categorical features of the parents. We assign

the same probabilistic importance to each parent. At the end of the crossover, we obtain anew a population of size L (half parents, half offsprings).

Next, we apply mixed polynomial mutation to alter randomly the mutable features of the offspring (Line 13). Each feature has a probability p_m to be altered (set to $p_m = |\mathcal{F}|^{-1}$ in our experiments). Like the initialisation process of the population, the applied mutation operators take into account the nature (categorical/integer or real) and boundaries of each feature. At the end of the mutation process, we obtain a new population P_j to proceed in the next generation.

After the specified number of generations passed, the algorithm returns the examples of the last generation that satisfy the constraints. In addition to these individuals, the algorithm also returns the associated values of fitness and objective functions.

4.3.4 Experimental Setup

To address our research questions, we implemented CoEvA2. The tool was developed in Python on top of PyMoo, an established framework for modelling and executing genetic algorithms in Python. Our implementation is publicly available.[5] All experiments were run on our partner's internal server with about 6 cores allocated for our experiments.

We set the meta-parameters of the genetic algorithm as follows. Population size was set to 40 to maintain an acceptable computation time (CoEva2 run on 4,000 initial states takes about 24 days). Exploratory experiments showed that a higher population size does not affect our results. Also, we stop the algorithm after 10,000 generations. These numbers were found experimentally to be sufficient in making our technique to craft successful adversarial examples. For selection, mutation and crossover, we kept their default parameters which worked well in our case. During our experimentation, we performed exploratory trials with alternative settings and observed minor differences. This is in line with the study of Zamani and Hemmati [ZH19] on the sensitivity of search-based testing methods to their hyper-parameters.

All our experiments focus on our partner's case study, i.e. generating feasible, adversarial overdraft requests approved by the CSS. To that end, we consider our partner's real-world data comprising 400,000 requests. The 275,000 were used by our partner to train the CSS's random forest. Out of the 125,000 remaining (the test set), we keep only those which are rejected overdraft requests correctly classified by the CSS. The rationale is that in realistic settings, an attacker can only manipulate future transactions and account status (which are inherently outside the training set) with the aim to make previously-rejected requests accepted by the system and, doing so, retrieving money illicitly.

[5] https://github.com/UL-SnT-Serval/coeva2/tree/fse

This leaves us with 19,274 data points. We use two random samples of this set, each of which contains 4,000 initial states (customer account and transaction history): the first sample is used in RQ1 and RQ2 while the second is used to assess the adversarial training in RQ3. Thus, for each RQ we execute CoEvA2 4,000 times, once on each initial state. This is sufficient to rule out random effects (read more in Section4.7).

Here it must be noted, that the above settings are common to all RQs we investigate. Still the related settings required to answer each specific RQ are given at the beginning of the result Sections, i.e., those that answer RQ1 and RQ2 (Sections 4.4, 4.5, 4.6).

4.4 RQ1: Constrained Papernot and Random Search

Our first series of experiments consider (1) the Papernot attack extended to consider the domain constraint and (2) a random search that only considers the satisfaction of the constraints as objective (aka CoEvA2 with the same meta-parameters but using only f_4 as the fitness function). We regard these two attacks as baseline methods that we seek to improve.

Our extension of the Papernot attack differs from the original in three ways. First, it avoids visiting the nodes related to immutable features (thus, it never changes these features). Second, it checks the satisfaction of boundary constraints on the fly, each time a feature is altered. Third, it attempts to satisfy the other constraints by updating the dependent features.

To allow for fine-grained analysis of their results, we define four objective indicators. Each indicator reports the percentage of initial states from which a given method can produce a valid adversarial example. The objective corresponding to these indicators are:

O1: satisfy the domain constraints
O2: cause misclassification
O3: satisfy O1 and O2
O4: satisfy O3 and create a relevant overdraft (more than 1,000 currency units)

We evaluate the two baseline methods on a sample of 4,000 initial states (randomly picked from 19,274 rejected overdrafts). That is, we run each method 4,000 times (once per sampled initial state).

Results are shown in Table 4.3. Interestingly, none of the generated adversarial examples (by any of the two attacks) are valid (none of them satisfy **O4**). In the case of Papernot, a small number of the generated examples satisfy the domain constraints and about one-fourth overall cause misclassification. However, there is none that fulfil both objectives. This shows that straightforward extensions to unconstrained attacks (to make them consider the constraints) remain ineffective.

Table 4.3: Objective indicators of random search and constrained Papernot attacks

	Success rate	
Objective	Random search	Papernot
Constraints (O1)	**0.00%**	0.20%
Misclassification (O2)	57.15%	25.85%
O1 and O2 (O3)	**0.00%**	**0.00%**
O3 and overdraft amount (O4)	**0.00%**	**0.00%**

Table 4.4: Objective indicators achieved by Random search (f_4) and CoEvA2, using different fitness functions: All, (f_1, f_3, f_4), and (f_1, f_2, f_4). We compare the following objectives: Constraints satisfaction (O1), misclassification (O2), constraints satisfaction and misclassification (O3 = O1 + O2), and constraints satisfaction, misclassification and overdraft amount maximization (O4)

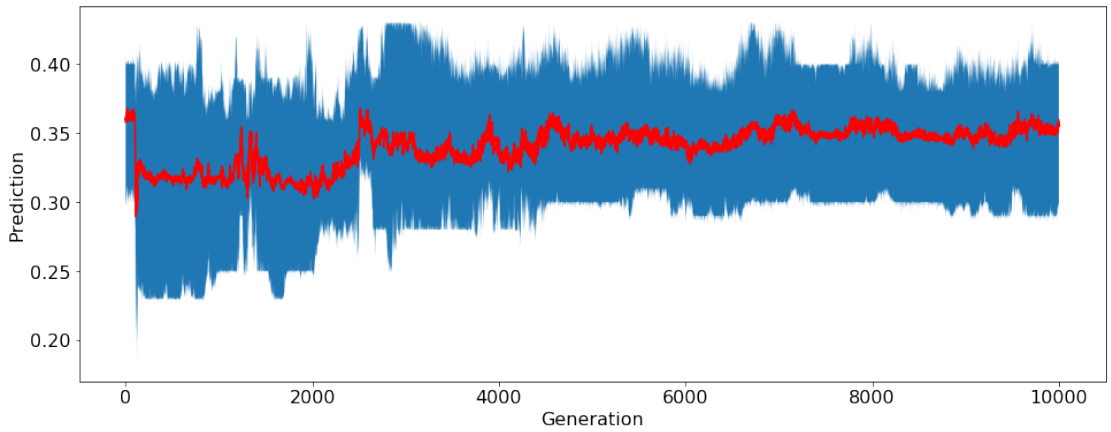
Objective indicators	Random search	All	f_1, f_3, f_4	f_1, f_2, f_4
Constraints (O1)	0.00%	58.9%	0.00%	100.00%
Misclassification (O2)	57.15%	27.1%	31.18%	18.79%
O1 and O2 (O3)	0.00%	17.2%	0.00%	18.79%
O3 and overdraft (O4)	0.00%	**8.45%**	0.00%	0.00%

In the case of the random search, we observe that more than half of the returned examples cause misclassification. Interestingly, none of them satisfy the constraints although this is the only objective forced upon the search. A detailed investigation of the generated examples reveals that the perturbation amount ranges from 0.2 to more than 1,000. This is significantly more than the Papernot attack and the aforementioned gradient-based methods (see our preliminary study Section 4.2.4). From these observations, we hypothesize that minimizing the perturbation would allow restricting the exploration within a reasonable area around the initial state. Doing so, the search would increase the likelihood to find valid adversarial examples around this initial state (in particular, when initializing the population and performing mutation).

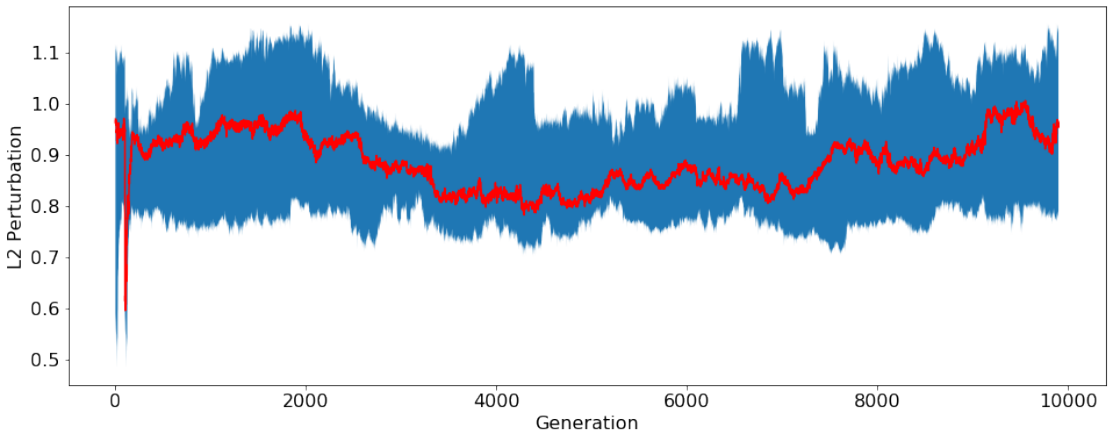
4.5 RQ2: CoEvA2 and its Fitness Function

Given that the baseline methods do not generate valid adversarial examples, we implement and evaluate CoEvA2. We execute the algorithm on the same randomly-picked set of the 4,000 initial states that was used in RQ1. We also consider the same four objective indicators as in RQ1 to allow for fine-grained analysis.

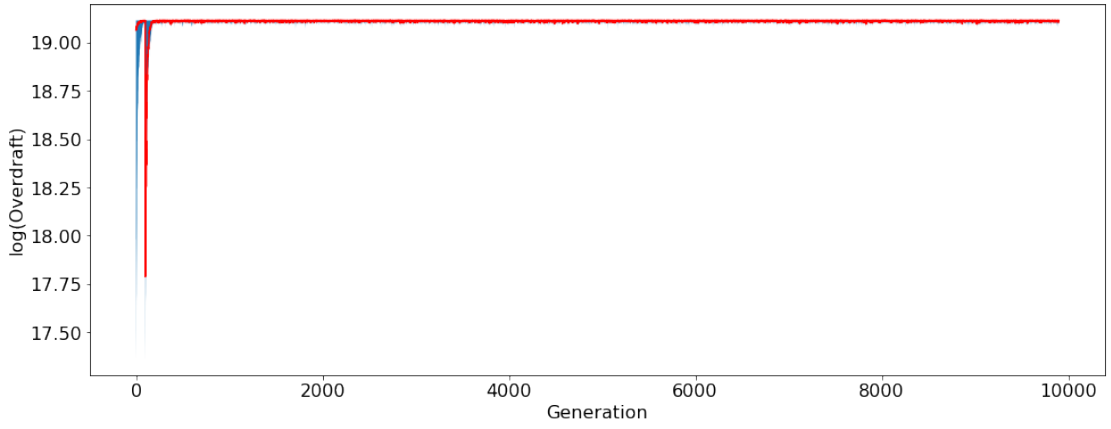
To identify and form a good fitness function, we consider multiple variants



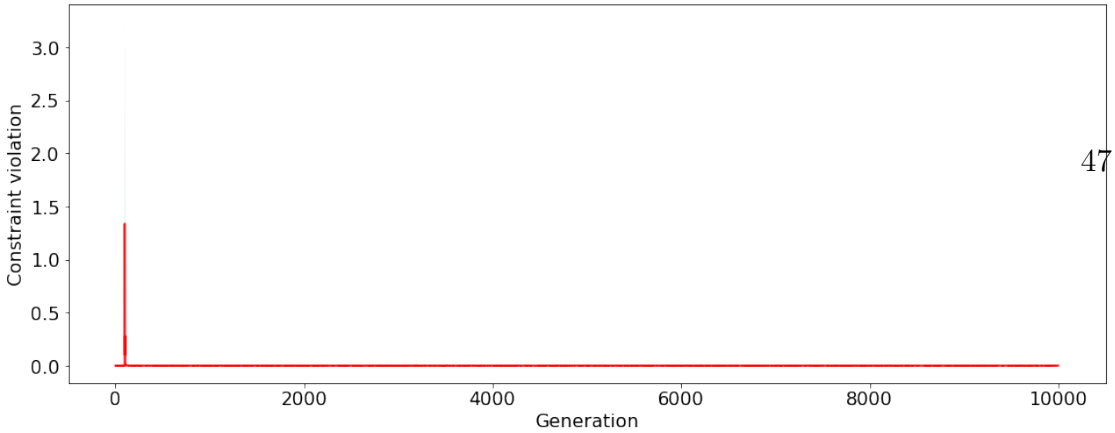
(a) f1: Prediction probability (lower is better)



(b) f2: Perturbation, L_2 distance (lower is better)



(c) f3: Overdraft amount (higher is better)



(d) f4: Constraints violation error (lower is better)

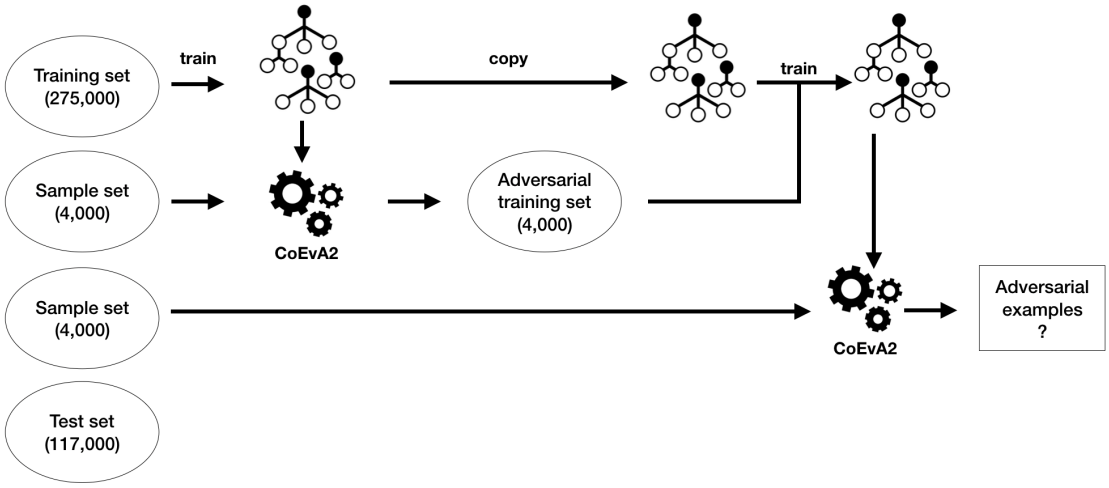


Figure 4.4: Adversarial training process.

of CoEvA2, each of which uses a different subset of the objective functions. In addition to the random search guided only by the constraint satisfaction (previously studied in RQ1), we consider three variants: the full CoEvA2, another variant where only the f_2 (perturbation minimization) part is removed and another one where only the f_3 (overdraft maximization) part is removed. Misclassification and constraint satisfaction are minimum mandatory criteria in order to generate valid examples and thus, all the three CoEvA2 variants we examine include them.

Table 4.4 summarizes our results. It shows that the variant of CoEvA2 with all parts of the objective function activated is the only one capable of generating adversarial examples that cause misclassification, satisfy the constraints and engender relevant overdrafts.

CoEvA2 is successful for 8.45% of the initial states. Thus, on average, only 12 initial states are needed to perform a successful attack. An interesting observation here is that from one initial state we can generate more than one valid adversarial example. This results in more than thirteen thousand of valid adversarial examples bypassing the banking system. These results seem to suggest that the fitness function we form is effective and that all its parts are important.

This last point can be confirmed by the rest of the recorded results. These show that all the objective function parts are necessary to generate successful adversarial examples. Without the perturbation minimization objective (one-before-last column), CoEvA2 generates slightly more misclassified examples (for 31.18% of the initial states instead of 27.10%) but none of them satisfies the constraints. This confirms our previous hypothesis that not restricting the perturbation makes the algorithm create examples much different from the original (valid) example. In highly constrained search space, this increases the likelihood of generating invalid

examples.

Finally, without the objective of maximizing the overdraft amount (last column), CoEvA2 generates examples satisfying the constraints in every case. However, only 18.79% of them cause misclassification. None of these achieve a sufficient overdraft amount (1,000 currency units). This is because the algorithm applies only small variations and only to the other mutable features, which reduces the likelihood of violating the constraints and of misclassification.

To better understand how CoEvA2 handles the trade-off between the four objective functions, we show in Figure 4.3 how the value of each of them evolves over the generations when applied to the initial states for which it managed to generate valid adversarial examples over 10,000 generations.

At each generation, we average the objective function scores obtained by the current population. Thus we obtain, for each initial state and each objective function, 10,000 values (one per generation). Then, we show the minimum, mean and maximum values of each (averaged) score over all the initial states. The red line is the mean, whereas the blue area denotes the minimal and maximal scores.

The four plots confirm that constraint satisfaction is the first objective fulfilled by the algorithm, and it does so always in the early generations (after about 100). The figure also shows that the overdraft amount is the second-most dominant objective and is always achieved within the first 200 generations, reaching 10^8 currency units, the maximum amount authorized by the CSS. All the individuals of the population in all the next generations inherit this maximum value and keep satisfying the constrains. Meanwhile, the L_2 distance fluctuates around 0.9, which is 10 times less than the Papernot attack. The average prediction probability stabilizes around 0.35, which is slightly below the prediction threshold.

Interestingly, taken together these results suggest that a careful choice of the initial state allows CoEvA2 to find valid adversarial examples after a limited number of generations (200). Moreover, these examples make the system overdraft of high amount (close to the strict maximum authorized by the bank). While frightening, these results also mean that we can focus on specific initial states to build countermeasures and increase the robustness of the system.

Overall, our results corroborate the conclusion that all four parts of our fitness function play a crucial role in crafting valid adversarial examples. At the same time, our approach demonstrates that it is indeed feasible to craft valid adversarial examples in real-world critical systems. This motivates the need for appropriate defence mechanisms to reinforce the robustness of the system against such attacks. We investigate this in the next research question.

4.6 RQ3: Adversarial Training

Figure 4.4 shows the adversarial training process we designed to improve the robustness of our partner's system against our (previously successful) adversarial attack. First, we generate 4,000 valid adversarial examples (overdrafts accepted by the model that should be rejected) and re-train the model to classify them correctly. To do so, we use the 4,000 initial states used in RQ2. After re-training the model, we execute again CoEvA2 4,000 times using each time a new initial state that was not used to produce the adversarial training set. We check whether the attack managed to generate any adversarial examples.

It results that the adversarial training makes CoEvA2 incapable of generating valid adversarial examples. Thus, our adversarial training method grants protection against the very same attack that was previously effective. This is a positive outcome that can be used by our partner in order to improve the robustness of the CSS. As our system is still in testing phase, it is used in parallel to the original model. Thus, an overdraft approved by the existing model and rejected by ours is likely to be an adversarial example. In all the other cases, one should follow the decision of the first model.

4.7 Threats to validity

Validity threats to our results may arise by the implementations we used. Thus, potential bugs either in our or the underlying frameworks may influence our results. We do not consider this threat as important since we thoroughly checked our code and many of the adversarial examples we generated were verified by our partner. Moreover, we rely on widely used and relatively reliable frameworks, Scikit-Learn and Tensorflow for the machine learning algorithms, and reputable libraries like the Adversarial Toolbox from IBM[6] for adversarial attacks and Pymoo from the Michigan State University [7] for multi-objective genetic algorithms.

Another potential threat concerns the specificity of the dataset and classification model we used. Both are from our partner's real production system and since our partner is a major player, its data and practices should be representative of other companies. Moreover, a verification of historical data revealed remarkable results for the last year. Due to the specificity of our industrial case, the results we obtained may not fully transfer to other industries (namely outside of credit scoring domain). Nevertheless, our endeavour shows that the problem exists in the real world and formalises it to facilitate the design of similar solutions to other cases. Moreover, our algorithm and approach have been designed to be generic enough to be adjusted to other use cases, and we provide the algorithm and all the

[6]https://github.com/IBM/adversarial-robustness-toolbox/
[7]https://pymoo.org

hyper-parameters of our approach for reproducibility.

To reduce the impact of random effects, all our experiments consider 4,000 different initial states (customer account and transaction history) and run the studied methods once per state. Since we make 4,000 independent executions, multiple runs per execution can only make a difference in isolated cases and not in the overall performance (expected case). This is because initial states can be seen as independent repetitions.

In our experiment, we perform a single run per state since we focus on trends. Thus, we run our approach on 4,000 cases and found adversarial cases in 341. These are sufficiently large numbers to rule out random effects. Yet, multiple repetitions and more generally additional search time-budget may improve the results of the search. We run the random method $4,000 \times 40,000$ times (4,000 initial states $\times$ 1,000 generations with 40 individuals), and found 0 adversarial cases, which demonstrates the ineffectiveness of the random method.

4.8 Conclusion

In this work, we studied the problem of testing a machine-learning-based industrial credit scoring system against malicious inputs. In particular, we considered the case where an attacker manipulates the related features, with the aim to cause a misclassification by a binary classification model. To this end, we evaluated the current state-of-the-art adversarial attacks, both in a full-knowledge context and a limited-knowledge using our partner's dataset and system. Based on this study, we shew that approaches proposed in the literature can indeed generate adversarial examples but these are not useful since they do account for domain constraints. This limitation of the methods results in generating implausible examples. To deal with this situation, we proposed a search-based method overcoming these limitations. We showed that our new attack constitutes a real security threat to FinTech systems relying on machine learning. At the same time, we exploit this threat to improve the defence mechanisms of our industrial system. In the end, the system becomes immune to our attack.

4.8.1 Artifact

Our library is available on Github and can easily be applied or extended to any constrained adversarial attack task. The repository contains multiple branches:

- **fse**: this branch tackles the implementation presented in this paper and the experiments to reproduce our results. Our dataset being proprietary and private, we provide a similar open-source dataset, *Lending Club Load Dataset*[8] to evaluate our approach.

[8]https://www.kaggle.com/wendykan/lending-club-loan-data

The tool is built around configuration files (located in */configurations* folder). JSON files where you can define the constraints of your problem, the weights of your objective functions, etc...
The folder */src* contains the actual implementation of our algorithm, while the folder */experiments* provides scripts to easily run each of the Research Questions' experiments.
An example of constrained dataset is provided in folder */data*. The constraints are plain python scripts that are loaded and evaluated dynamically.

In the provided dataset, one of the constraints is that the *Number of public record bankruptcies* should not exceed the *Number of derogatory public records*. A valid adversarial attack should therefore not break this logical constraint. More information and instructions are provided in the branch's README.md.

- **master**: this branch is the ongoing iteration of the library. It provides an extension of the approach using Grid and Random search to optimize the weights of each objective, and *MoEva2* an NSGA-2 extension of our approach that uses non-dominated multi-objective evolution. This branch will contain the stable evolutions of the library, in particular all elements mentioned as future or ongoing work.

5

Adversarial Robustness in Multi-Task Learning: Promises and Illusions.

Vulnerability to adversarial attacks is a well-known weakness of Deep Neural networks. While most of the studies focus on single-task neural networks with computer vision datasets, very little research has considered complex multi-task models that are common in real applications. In this paper, we evaluate the design choices that impact the robustness of multi-task deep learning networks. We provide evidence that blindly adding auxiliary tasks, or weighing the tasks provides a false sense of robustness. Thereby, we tone down the claim made by previous research and study the different factors which may affect robustness. In particular, we show that the choice of the task to incorporate in the loss function are important factors that can be leveraged to yield more robust models. We provide all our models, and open source-code at `https://github.com/yamizi/taskaugment`.

Contents

5.1 Introduction

While most research on computer vision focuses on single-task learning, many real applications (especially in robotics [RVB18], autonomous vehicle [YCW+20], privacy [GCP+19a] and medical diagnosis [XY11]) require learning to perform different tasks from the same inputs. For instance, an autonomous vehicle processes multiple computer vision tasks to navigate properly [SEZ+18; SKM+19], such as object segmentation and depth estimation.

To meet such requirements, one can build and train on a specific model for each task. However, previous studies have shown that multi-task learning, i.e. building models that learn to perform multiple tasks simultaneously, yields better performance than learning the individual tasks separately [ZY17; VGV+21]. The intuition behind these results is that tasks that share similar objectives benefit from the information learned by each other. However, while the performance of multi-task learning approaches on clean images has seen major improvements recently[SZC+20], the security of these models, in particular their vulnerability to adversarial attacks, has been barely studied.

The phenomena of *adversarial attacks* has first been introduced by [BCM+13] and [SZS+13] and has since gathered the interest of researchers to propose new attacks [GSS14a; MMS+17b], defense mechanisms [KGB16; HWC+17], detection mechanisms [MGF+17], or to improve transferability across different networks [TPG+17; IWL+19].

[MGN+20] pioneered the research on adversarial attacks against multi-task models. Their main result is that, under specific conditions, making models learn additional *auxiliary* tasks leads to both an increase in clean performances *and* improves the robustness of models.

In this paper, we pursue the endeavor of Mao et al. and study whether their conclusions hold in a larger spectrum of settings. Surprisingly, our experimental study shows that adding more tasks does not consistently increase robustness, and may even have negative effects. We even reveal some experimental parameters, such as the attack norm distance, can annihilate the validity of the previous theoretical results. Overall, this indicates that increasing the number of tasks only gives *a false sense of robustness*.

In face of this disillusionment, we investigate the different factors that may explain the discrepancies between our results and that of the previous study. We demonstrate that the contribution of each task to the vulnerability of the model can be qualified, and that what matters is not the number of added tasks but the *marginal vulnerability* of these tasks.

Following this finding, we investigate remedies to reinstate the addition of auxiliary tasks as an effective means of improving robustness. Firstly, following the recommendation of previous research on increasing the clean performance of

multi-task models [CBL+18; SZC+20], we show that adjusting the task weights to make vulnerable tasks less dominant yields drastic robustness improvement but does so only against non-adaptive adversarial attacks: Auto-PGD [CH20a] and Weighted Gradient Descent – a new adaptive attack that we propose to adapt the produced perturbation to task weights – annihilate the benefits of weight adjustment.

Secondly, we show that a careful selection of the tasks suffices to ensure that model robustness increases. Given a target model, determining which combination of tasks is optimal can be costly. We propose different methods to approximate the gain in robustness and show that they strongly correlate with the robustness of the target model.

To summarize, our contributions are:

- We show that adding auxiliary tasks is not a guarantee of improving robustness and identify the key factors explaining the discrepancies with the original study.
- We refine the theory of Mao et al. through the concept of *marginal adversarial vulnerability* of tasks. Leaning on this, we demonstrate that the inherent vulnerability of tasks plays a central role in the model robustness.
- We empirically show that a careful weighting of the tasks can act as a remedy and offer the benefits initially promised by previous research. However, it does not provide increased robustness against adaptive attacks.
- We propose a set of surrogates to efficiently evaluate the robustness of a combination of tasks.

5.2 Problem Formulation

5.2.1 Preliminaries

Let $\mathscr{M}$ a multi-task model with tasks $\mathscr{T} = \{t_1, ..., t_M\}$. For each input example x, we denote by $\bar{y}$ the corresponding ground-truth label and we have $\bar{y} = (y_1, ..., y_i, y_M)$ where y_i is the corresponding ground truth for task i.

We focus on hard sharing multi-task learning, where the models are made of one encoder (backbone network common to all tasks) $E(.)$ and task-specific decoders $D_i(.)$. Each task is associated with a loss $\mathscr{L}_i$; $\mathscr{L}_i(x, y_i) = l_i(D_i(E(x)), y_i)$ where l_i is a loss function tailored to the task i. For instance, we can use cross-entropy losses for classification tasks, and L1 losses for regression tasks.

The total loss $\mathscr{L}$ of our multi-task model is a weighted sum of the individual losses $\mathscr{L}_i$ of each task:

$$\mathscr{L}(x, \bar{y}) = \sum_{i=1}^{M} w_i \mathscr{L}_i(x, y_i)$$

where $\{w_1 \dots w_M\}$ are the weights of the tasks, either set manually or optimized

during training [CBL+18].

Single-task adversarial attacks In this use case, the adversary tries to increase the error of one single task. This threat model can represent scenarios where the attacker has access to one task only or aims to perturb one identified task. The objective of the attacker can then be modeled as:

$$\underset{\delta}{\operatorname{argmax}} \mathscr{L}_i(x+\delta, y_i) \text{ s.t. } \|\delta\| p \leq \epsilon \tag{5.1}$$

where i is the index of the targeted task and ϵ the maximum perturbation size using a norm p.

Multi-task adversarial attacks In multi-task adversarial attacks, the adversary aims to increase the error of multiple outputs all at once. This captures scenarios where the adversary does not have fine-grained access to the individual tasks or where the final prediction of the system results from the combination of multiple tasks. Therefore, the adversary has to attack all tasks together.

Given a multi-task model $\mathscr{M}$, an input example x , and $\bar{y}$ the corresponding ground-truth label, the attacker seeks the perturbation δ that will maximize the joint loss function of the attacked tasks – i.e. the summed loss, within a p-norm bounded distance ϵ.

The objective of the attack is then:

$$\underset{\delta}{\operatorname{argmax}} \mathscr{L}(x+\delta, \bar{y}) \text{ s.t. } \|\delta\| p \leq \epsilon \tag{5.2}$$

Adversarial vulnerability of multi-task models [SOB+19] introduced the concept of adversarial vulnerability to evaluate and compare the robustness of single-task models and settings. Mao et al. extended it to multi-task models as follow:

Definition 2. *Let $\mathscr{M}$ be a multi-task model. $\mathscr{T}' \subseteq \mathscr{T}$ a subset of its tasks and $\mathcal{L}_{\mathscr{T}'}$ the joint loss of tasks in $\mathscr{T}'$. Then, we denote by $\mathbb{E}_x[\delta\mathcal{L}(\mathscr{T}', \epsilon)]$ the* adversarial vulnerability *of $\mathscr{M}$ on $\mathscr{T}'$ to an ϵ-sized $\|.\|_p$-attack, and define it as the average increase of $\mathcal{L}_{\mathscr{T}'}$ after attack over the whole dataset:*

$$\mathbb{E}_x[\delta\mathcal{L}(\mathscr{T}', \epsilon)] = \mathbb{E}_x\left[\max_{\|\delta\|_p \leq \epsilon} \mid \mathcal{L}_{\mathscr{T}'}(x+\delta, \bar{y}) - \mathcal{L}_{\mathscr{T}'}(x, \bar{y}) \mid\right]$$

This definition matches the definitions of previous work [GSS14b; SNV+17] of the robustness of deep learning models: the models are considered vulnerable when a small perturbation causes a large average variation of the joint loss.

Similarly, we call *adversarial task vulnerability* of a task i the average increase of $\mathcal{L}_{\mathscr{T}'}(x, y_i)$ after an attack.

Assuming that the variation δ is small, [MGN+20] proposed the following theorem using the [SOB+19] first-order Taylor expansion in ϵ:

Theorem 1. *Consider a multi-task model $\mathscr{M}$ where an attacker targets $\mathscr{T} = \{t_1, ..., t_M\}$ tasks uniformly weighted, with an ϵ-sized $\|.\|_p$-attack. If the model is converged, and the gradient for each task is i.i.d. with zero mean and the tasks are correlated, the adversarial vulnerability of the model can be approximated as*

$$\mathbb{E}_x[\delta\mathcal{L}'] \approx K \cdot \sqrt{\frac{1 + \frac{2}{M}\sum_{i=1}^{M}\sum_{j=1}^{i-1}\frac{\mathrm{Cov}\left(\mathbf{r}_i, \mathbf{r}_j\right)}{\mathrm{Cov}(\mathbf{r}_i, \mathbf{r}_i)}}{M}} \tag{5.3}$$

where K is a constant dependant of ϵ and the attacked tasks, and $\mathbf{r}_i = \partial_x \mathcal{L}(x, y_i)$ the gradient of the task i, and $\mathrm{Cov}\left(\mathbf{r}_i, \mathbf{r}_j\right)$ the covariance between the two gradients $\mathbf{r}_i, \mathbf{r}_j$.

Proof. Appendix A.1 and A.2 □

This theorem indicates that under the specific assumptions described above, (1) increasing the number of tasks reduces the adversarial vulnerability of a multi-task model and (2) even more when these tasks are uncorrelated.

5.2.2 Research Questions and Methodology

Our research endeavor stems from the hypothesis that the assumptions supporting the above theorem are too restrictive to be met in practice. The existence of settings where these assumptions do not hold would tone down the validity of the results of [MGN+20] and raise anew the question of how to achieve robust multi-task learning.

Accordingly, our first research question investigates if the assumptions and results of Theorem 1 are confirmed in a variety of multi-task models and settings. We ask:

RQ1: *Are multi-task models reliably more robust than single-task models?*

To answer this question, we study whether the results of Theorem 1 (i.e. adding tasks increases robustness) generalize to other settings. In particular, Theorem 1 was proven for l_2-normed attacks and we investigate if their results remain valid for l_∞-normed attacks. Finally, we investigate the impact of various experimental settings covering different perturbation budgets ϵ, architectures, and number of training steps.

Following this, we formulate an alternate hypothesis that could explain the apparent robustness of multi-task models and the evidence brought by our study:

what matters most is not the number of tasks or how they correlate but how much the tasks individually impact the vulnerability of the model. Thus, making more robust a model with one task that is "marginally more vulnerable" requires adding robust tasks that make this model "marginally less vulnerable".

Our second research question studies this hypothesis and attempts to quantify this *marginal vulnerability* :

RQ2: *How to quantify the individual contribution of each task on the robustness of the model?*

To answer this question, we define the concept of marginal adversarial vulnerability of a model to a task i as the variation between the adversarial vulnerability of the model with this newly added task i and its vulnerability before this task.

Based on this concept of marginal adversarial vulnerability, we look for ways to improve robustness. Past research on multi-task learning suggests that carefully weighing tasks can drastically improve clean performance [VGV+21]. Leaning on this idea, we hypothesize that one can improve model robustness by adjusting the weights of the tasks. We ask:

RQ3: *Can one improve the robustness of multi-tasks models by adjusting the weights of the tasks?*

We show that optimizing the weights significantly improves the robustness of multi-tasks models against PGD. However, this apparent robustness may actually result from a gradient masking effect [ACW18] caused by the weight adjustment. We, therefore, investigate if adaptive attacks – which are known to circumvent gradient masking – can successfully attack the weight-optimized model. We use Auto-PGD [CH20a] and propose a new attack that adjusts at each step of the attack which tasks are attacked and how much the gradient of each task is penalized to compute the optimal perturbation.

Finally, we investigate the practical question of how to identify the combinations of tasks that yield the highest robustness. Given two multi-task models, we propose a set of guidelines that help practitioners to infer the robustness of the task combinations of each model from cheaper models. Our final question is:

RQ4: *How to efficiently find combinations of tasks giving the best robustness?*

5.2.3 Experimental Setup

The following describes our general experimental setup used across all RQs. It must be noted that the setups specific to the RQs are presented in their dedicated sections.

Dataset. We use the Taskonomy dataset, an established dataset for multi-task learning [ZSS+18]. From the original paper, we focus on 11 tasks : Semantic Segmentation (s), Depth z-buffer Estimation (d), Depth euclidian Estimation (D), Surface Normal Prediction (n), SURF Keypoint Detection in 2D (k) and 3D (K), Canny Edge Detection (e), Edge Occlusion (E), Principal Curvature (p), Reshading (r) and Auto-Encoders (A). This subset of tasks is at the intersection of the major studies [ZSS+18; VGD+19; SZC+20] about tasks similarity.

Attacks. We focus our research on gradient-based attacks. In particular, we use as base setting the l_∞ Projected Gradient Descent attack (PGD) [MMS+17b] with 25 steps attacks, a strength of $\epsilon = 8/255$ and a step size $\alpha = 2/255$.

We also study in RQ3 the impact of adaptive attacks on the robustness of multi-task models. We evaluate the robustness of weighted multi-task models against Auto-PGD [CH20a], the strongest parameter-free gradient attack.

Finally, we design a new attack, *WGD*, that takes into account individual task weights and show that multi-task learning is vulnerable against our adaptive attack.

Models. We use the architectures and training settings of the original Taskonomy paper [ZSS+18]: A Resnet18 encoder and a custom decoder for each task.We use a uniform weights, Cross-entropy loss for the semantic segmentation task and an L1 loss for the other tasks.

Our evaluation covers all possible combinations of tasks, i.e 935 multi-task models. Each model is a combination of a main task and one or multiple auxiliary tasks. We present in our evaluation the case where the attacker aims to attack only the main task (single-task attacks) and when all the tasks are attacked simultaneously (multi-task attacks).

We provide in the Appendix B. of supplementary material the detailed setup of each setting and the detailed results.

5.2.4 Robustness Metrics

Task robustness The common way to evaluate empirically the adversarial robustness of a DNN is to compute the success rate of the attack, i.e. the percentage of inputs for which the attack can produce a successful adversarial example under a constrained perturbation budget ϵ.

While this metric is suited for classification tasks that rely on classification accuracy, most dense tasks rely on metrics where a success rate is hard to define objectively or requires a hand-picked threshold. For instance, "image segmentation" uses Intersection Over Union (IoU, between 0 and 1) as a metric, while "pose estimation" relies on the number of correct keypoints and their orientation, and "depth estimation" uses the mean square error. To account for this diversity of metrics we define a generic metric that reflects how much the performance has degraded (how much relative error) after an attack: *the relative task vulnerability* metric.

Relative task vulnerability Given a model $\mathscr{M}$, we define the relative task vulnerability v_i of a task i as the average relative error increase when changing clean inputs $x^{(k)}$ into adversarial inputs $x^{(k)} + \delta$, given their associated ground truth $y_i^{(k)}$. Hence, v_i is given by:

$$v_i = \mathbb{E}_k \left[\frac{f_i(x^{(k)} + \delta, y_i^{(k)}) - f_i(x^{(k)}, y_i^{(k)})}{f_i(x^{(k)}, y_i^{(k)})} \right]$$

where error function $f_i(x^{(k)}, y_i^{(k)})$ is a task-specific error between the ground truth $y_i^{(k)}$ and the predicted value of $x^{(k)}$ (e.g., MSE, 1-IoU, etc.).

The concept of relative task vulnerability enables the comparison of two models in terms of the robustness they achieve for any of their task(s). A model with a smaller value of v_i indicates that it is more robust to an attack against task i.

5.3 RQ1: Adding Auxiliary Tasks

Theoreom 1 showed that adversarial vulnerability decreases with the number of uncorrelated tasks. We argue that the results do not generalize to different settings (norms, attack strength, ...) and investigate the other factors which may affect the robustness of the models. This allows us to identify the key factors confirming or refuting the results of Theorem3. More precisely, we consider the attack norm p, the perturbation budget ϵ, the model architecture, and the number of training steps (the convergence of learning). We summarize our findings below while detailed results are in Appendix C.

Attack norm p. We evaluate the relative task vulnerability against single-task and multi-task attacks with a limited perturbation budget ($\epsilon = 8/255$, 25 attack steps) under l_2 and l_∞ attacks. Under l_2 attacks, the previous conclusions of [MGN+20] are confirmed. Under l_∞ attacks, however, adding auxiliary tasks does not reliably increase robustness against neither single-task nor multi-task adversarial attacks. We indeed observe for each task t that the single-task model of t is not more vulnerable than multi-task models with t as the main task – regardless of the fact that a single-task attack or a multi-task attack was used (see, e.g., also Table 5.1).

Attack budget ϵ. We evaluate the robustness of multitask models against attack size $\epsilon \in \{4/255, 8/255, 16/255, 32/255\}$ Under strong adversarial attacks ($\epsilon > 4/255$), multi-task learning does not provide reliable robustness both against single-task and multi-task adversarial attacks.

Model architecture. We evaluate the vulnerability of multi-task models on three families of encoders (Xception, Wide-Resnet, and Resnet) and for the latter, three sizes of encoders (Resnet18, Resnet50 and Resnet152). We train each architecture

on a pair of tasks from s, d, D, E, n. We evaluate the robustness of each combination of tasks and compare it with the robustness of the same architecture trained using only one of the tasks. For all architectures, multi-task models are not reliably less vulnerable than single-task models. On the contrary, when one task is targeted, 80% of the multi-task models using Resnet50 and Resnet152 architectures are more vulnerable than their single-task counterparts.

Answer to RQ1: For large perturbation budgets ϵ, l_∞ norms, or large models, multi-task learning does not reliably improve the robustness against adversarial attacks.

5.4 RQ2: Marginal Adversarial Vulnerability

5.4.1 Theoretical Analysis

To better understand the impact of additional tasks on the multi-task vulnerability, we define the concept of marginal adversarial vulnerability of tasks, and propose a new theorem that bounds the contribution of the additional tasks to the model's vulnerability.

Definition 3. *Let $\mathscr{M}$ be a multi-task model with $\mathscr{T} = \{t_1, ..., t_M\}$ tasks, an input x, $\bar{y} = (y_1, ..., y_M)$ its corresponding ground-truth. We denote the set of attacked tasks $\mathscr{T}_N$ and $\mathscr{T}_{N+1}$, two subsets of the model's tasks $\mathscr{T}$ such as $\mathscr{T}_{N+1} = \mathscr{T}_N \cup \{t_{N+1}\}$ and $N + 1 \leq M$, and let $\mathcal{L}'$ be the joint task loss of attacked tasks.*

We define marginal adversarial vulnerability *of the model to an ϵ-sized $\|.\|_p$-attack as the difference between the adversarial vulnerability over the task set $\mathscr{T}_{N+1}$ and the adversarial vulnerability over the task set $\mathscr{T}_N$. It is given by:*

$$\Delta_N \mathbb{E}_x[\delta\mathcal{L}'] = \mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_{N+1})] - \mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_N)]\Big]$$

Similarly to the adversarial vulnerability of a model, for a small δ value we propose to use Taylor expansions to approximate the marginal vulnerability of the model to a given task. We propose the following theory for the marginal change of the vulnerability of a model when we add a task:

Theorem 2. *For a given multi-task model $\mathscr{M}$, let $\mathbf{r}_i = \partial_x \mathcal{L}(x, y_i)$ the gradient of the task i, with a weight w_i and zero mean such as the joint gradient of $\mathscr{M}$ is defined as $\partial_x \mathcal{L}(x, \bar{y}) = \sum_{i=1}^{M} w_i \mathbf{r}_i$. The first-order approximation of the marginal vulnerability is bounded as follow:*

$$\widetilde{\Delta_N \mathbb{E}_x[\delta\mathcal{L}']} \leq \epsilon \cdot ((N+1) \cdot w_{N+1} \mathbb{E}_x[|| \mathbf{r}_{N+1} ||] + N \cdot \max_{i<N+1} w_i \mathbb{E}_x[|| \mathbf{r}_i ||])$$

Proof. Appendix A.4 and A.5 □

When all the tasks have the same weight, $w_i = \frac{1}{N}$ and $w_{N+1} = \frac{1}{N+1}$ and we have:

$$\widetilde{\Delta_N \mathbb{E}_x[\delta \mathcal{L}']} \leq \epsilon \cdot (\mathbb{E}_x[|| \mathbf{r}_{N+1} ||] + \max_{i<N+1} \mathbb{E}_x[|| \mathbf{r}_i ||])$$

This theorem shows that the increase in adversarial vulnerability when we add more tasks does not depend on the number of tasks already attacked but relates to how robust is the new task we are adding and how weak is the most vulnerable task of the model.

5.4.2 Empirical Evaluation

To confirm our hypothesis that increasing the number of tasks does not guarantee the increase of robustness, we empirically evaluate how the adversarial vulnerability of a model changes when successively adding more tasks (up to 5 tasks). The model is trained with all 5 tasks then we successively enable the tasks. We show below the results for four combinations. Other combinations of tasks are provided in Appendix C.

Figure 5.1 shows the results. We observe that increasing the number of tasks often increases the vulnerability of the model. This confirms again that the main claim of [MGN+20] does not generalize to any combinations of tasks. We also observe that there is no monotonic relationship between the model vulnerability and its number of tasks for cases (2) and (3), whereas in cases (1) and (4) the increase is not linear and mostly occurs at one specific point (i.e., when the segmentation task s is added). More generally, across all four cases, task s appears to be the main factor contributing to the increased vulnerability of the model. This supports our claim that the most marginally vulnerable tasks are the dominant factors to increasing the model vulnerability.

Answer to RQ2: The marginal vulnerability increase of the model mainly depends on the vulnerability of the newly added task and the most vulnerable previous task. This implies that, (1) the more vulnerable the tasks in the model are, the less likely adding new tasks increases the robustness of the model; and (2) adding a vulnerable task may actually decrease the robustness of the whole model.

5.5 RQ3: Task Weight Optimization

We evaluate how the optimization of weights of the losses of each task can be used as a defense, through the selection of optimal weights. We investigate and as an adaptive attack to overcome the gradient masking of multi-task learning.

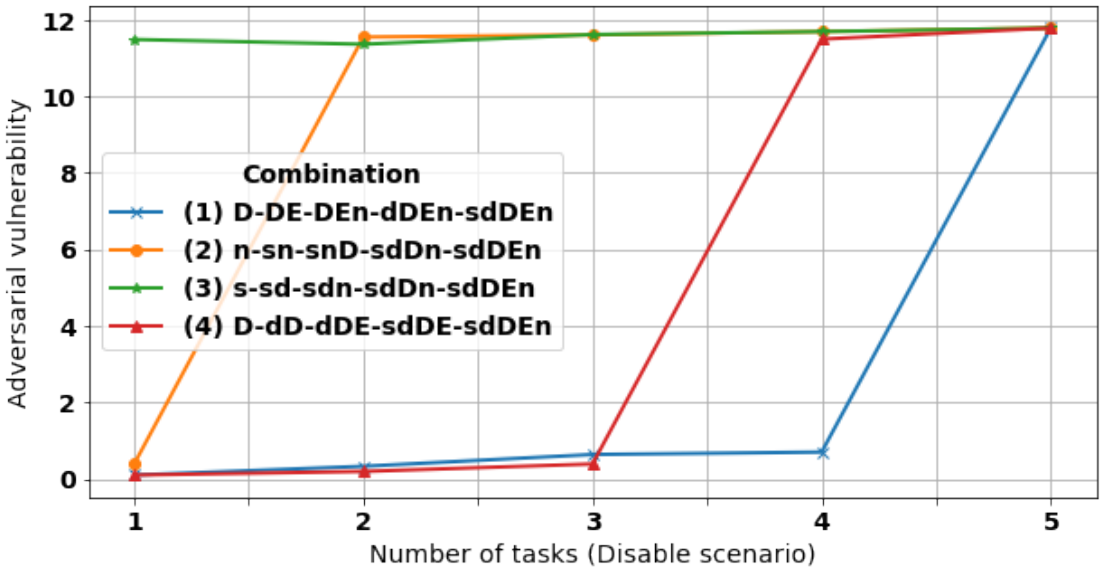


Figure 5.1: Adversarial vulnerability for 4 different combinations of tasks. In each combination, we enable one additional task and report the exact adversarial vulnerability of the new model. Evaluated tasks: *s*: Semantic segmentation, *d*: Z-depth, *D*: Euclidian depth, *n*: Normal estimation, *E*:Edge detection.

Attack		Baseline (A)					Weighted (B)				
Auxiliary →		s	d	D	n	E	s	d	D	n	E
Single	s	*0.82*	0.86	0.97	0.96	0.93	-	0.52	0.57	0.58	0.54
	d	5.74	*5.61*	5.28	6.88	6.41	1.25	-	2.00	1.65	1.92
	D	5.93	6.14	*6.4*	7.12	8.31	2.16	1.88	-	2.11	1.78
	n	7.43	9.48	8.93	*10.82*	9.08	6.61	9.46	8.09	-	8.14
	E	12.93	19.29	18.44	15.16	*22.57*	6.44	5.50	8.02	5.49	-
Multi	s	-	0.85	0.96	0.95	0.91	-	0.45	0.49	0.48	0.43
	d	1.99	-	5.42	4.8	4.46	0.56	-	2.06	0.82	0.75
	D	2.14	6.02	-	5.02	6.07	0.88	1.74	-	0.94	0.68
	n	4.61	9.4	8.93	-	8.7	3.65	9.22	7.99	-	6.32
	E	7.58	18.5	17.44	12.48	-	3.24	4.93	7.00	3.36	-

Table 5.1: Relative task vulnerability (lower is better). (A): adversarial attack against uniformly weighted tasks. (B): Adversarial attack against weighted tasks. Each row is the main task evaluated and the column is the auxiliary task. In the top half (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

5.5.1 Robustification Through Optimal Weights

Given that simply adding tasks is not a promise of increased robustness, we suggest that a better way would be to adjust the weights of the tasks (in the loss function). The problem of setting the task weights has been previously studied in the context of optimizing the clean performance of multi-task models [CBL+18; VGV+21; SZC+20]. Specifically, Zamir et al. [ZSS+18] provided the optimal weights for all combinations of tasks of their dataset (see Appendix B.).

To evaluate the potential benefits of adjusting the weights, we conduct an empirical evaluation and compare the vulnerability of models using equal task weights (the *Baseline* models) with the equivalent model using the optimized task weights suggested by [ZSS+18] (the *Weighted* models). More precisely, we consider any pair of tasks where the first task is the main task and the second is the auxiliary task (added for the specific purpose of making the model less vulnerable on the main task). To compare the equal-weight model with the optimized-weight model, we use the relative task vulnerability metrics as this metric is independent of the task weights. We use both single-task PGD [MMS+17b] and multi-task PGD [MGN+20].

Results are shown in Table 5.1 (Baseline A vs Weighted B). We see that for each pair of tasks, the weighted model is less vulnerable than the baseline model. This confirms our hypothesis that a careful setting of the weights can reduce model vulnerability, even where the addition of tasks with equal weights has a negative effect. For instance, in the baseline models, we observe that the vulnerability of the model on task s is lower when s is the only task than when any other task is added. When weights are optimized though, the vulnerability of the weighted multi-task model on s is always lower than s alone *regardless of the task that is added*. These results can be explained by the fact that the weight optimization proposed in [ZSS+18] aims to reduce the influence of dominant tasks during learning. As the two attacks inherently target the most dominant tasks (which have a higher average contribution to the loss function), the weight optimization improves both clean performance *and* robustness.

5.5.2 Adaptive Gradient Attacks

We investigate whether weight optimization remains an effective defense against adaptive attacks. We consider Auto-PGD [CH20a] – the strongest adaptive gradient attack in the literature – which adjust the step of the attack and the weight of the momentum at each attack iteration.

We also propose another way to make an attack adaptive. The principle of our new attack is to weight the contribution of each task when computing the perturbation to guarantee that the attack affects all targeted tasks – including those that have a smaller contribution to the joint loss.

We introduce the concept of Task Attack Rate to optimally weigh the gradient of each attacked task. Task Attack Rate is inspired by the inverse task learning rate proposed by [CBL+18] for the GradNorm optimization for training.

Definition 4. *We define the* Inverse task attack rate *of the task i under an ϵ-sized $\|.\|_p$- iterative attack on $\mathscr{T}'$ at step t the loss ratio for a specific task i at step t:* $\tilde{\mathcal{L}}_i(t) = \frac{\mathcal{L}_i(t)}{\mathcal{L}_i(0)}$

Smaller $\tilde{\mathcal{L}}_i$ implies that task i is faster to perturb. Similarly, we define the relative inverse attack rate as $r_i(t) = \frac{\tilde{\mathcal{L}}_i(t)}{\mathbb{E}_i[\tilde{\mathcal{L}}_i(t)]}$

We leverage this optimization in our new attack, *Multi-task Weighted Gradient Attack (WGD)*, a multi-step attack where the gradient of each task is weighted by the relative inverse task attack rate. We describe full algorithm in Appendix D.

We empirically assess whether the two adaptive attacks can overcome the robustification mechanism based on optimal weights. Hence, we measure the relative task vulnerability of the baseline (uniformly weighted) model and the optimally weighted model against WGD and Auto-PGD (Table 5.2).

We observe that, on both models, WGD and Auto-PGD are much stronger than PGD (compared to Table 5.1). For instance, Auto-PGD increased the error against task E up to five times in comparison with PGD on the same combination of tasks, and WGD caused up to two times more error than PGD for the combination of tasks supporting n.

Table 5.2 also reveals that the weighted model is as vulnerable as the baseline model. This confirms that the adaptive attacks negate the benevolent effects of weight optimization.

In the end, the only viable way to improve model robustness remains to add less vulnerable auxiliary tasks. Indeed, in Table 5.2 we observe that for each single-task model (i.e. the diagonal elements) there is at least one multi-task model (with the same main task) that is less vulnerable. Our previous (RQ2) conclusion remains, therefore, valid.

Answer to RQ3: Weight optimization in multi-task learning decreases model vulnerability against non-adaptive attacks only. The only way to improve the robustness of multi-task models remains to add less vulnerable auxiliary tasks.

5.6 RQ4: Task Selection

Our previous results imply that one should carefully select the auxiliary tasks added to reduce model vulnerability. Generally speaking, the addition of auxiliary tasks can even have negative effects. Auxiliary task selection, however, comes with three drawbacks [SZC+20]: the size of the model is bigger (due to the addition of the task-specific decoder), the convergence of the common encoder layers is slower,

Attack		Baseline (A)					Weighted (B)				
Auxiliary →		s	d	D	n	E	s	d	D	n	E
APGD	s	0.89	0.91	0.90	0.88	0.92	0.89	0.92	0.93	0.88	0.90
	d	17.17	23.88	13.50	24.10	24.98	18.27	23.19	13.08	15.92	23.9
	D	15.50	15.08	20.15	26.00	22.74	20.72	24.93	18.29	28.21	23.68
	n	12.99	17.76	17.27	19.02	17.24	12.35	17.14	16.72	18.49	16.46
	E	135.4	171.8	159.7	138.8	81.77	125.8	78.65	377.8	110.4	68.06
WGD	s	0.90	0.91	0.91	0.90	0.91	0.90	0.93	0.94	0.93	0.91
	d	12.86	13.57	12.39	16.18	18.13	12.66	13.55	12.8	11.7	12.96
	D	14.05	14.04	15.57	17.03	19.24	15.65	14.85	15.55	16.95	13.56
	n	13.05	17.06	16.32	18.12	16.57	17.00	20.26	17.32	18.13	17.68
	E	45.35	90.67	86.04	57.43	90.19	106.9	89.89	116.3	71.42	90.59

Table 5.2: Relative task vulnerability under two different attacks (lower is more robust). (A): adversarial attack against uniformly weighted tasks. (B): Adversarial attack against optimally weighted tasks. Each row refers to the task attacked and evaluated and the column the auxiliary task.

and the clean performance risk deteriorating as more tasks are added. This raises the question of how to select the combination that yields the lowest vulnerability *without* evaluating the vulnerability of all the possible combinations.

We propose three methods to make this selection more efficient. Their common idea is to compute a proxy of the adversarial vulnerability which is fast to get and correlated to the real adversarial vulnerability. We use the relative task vulnerability of the models on the main task, as this metric is independent of the task (unlike, the task loss whose scale depends on the task).

We use the Pearson coefficient to measure correlation between variables. We compute the correlation between the relative task vulnerability of one combination of tasks in the expensive model with the relative task vulnerability of the same combination on the cheaper surrogate.

The first method is *early stopping*, that is, training the model after a predefined (small) number of epochs. Here, we stop after 50 epochs while the full training lasts for 150 epochs. Strong correlations between the vulnerability of the models would indicate that one can decide which task combination is optimal after few epochs.

The second is to use a surrogate (less expensive) encoder and evaluate all task combinations on this encoder. We hypothesize that combinations of tasks that are effective when joint to the surrogate encoder are also effective with the original model. We use ResNet18 as the surrogate encoder and ResNet50 as the target encoder.

Our third selection method is guided by the clean performance on the main

task when the auxiliary tasks are added. The existence of a correlation between clean performance and vulnerability would allow avoiding the cost of evaluating adversarial vulnerability (e.g. applying PGD) and reuse existing results on clean performance predictions [SZC+20] to predict adversarial vulnerability.

Table 5.3 shows the Pearson correlation coefficient with the associated p-value. We observe that all three proxy methods are correlated with the real adversarial vulnerability values. Specifically, early stopping offers a medium correlation (0.55) while the methods based on the surrogate encoder and the clean performance achieve a very strong correlation (0.94).

Answer to RQ4: While exhaustively computing the adversarial vulnerability for all task combinations is computationally expensive, guiding the auxiliary task selection by the clean performance or the vulnerability of a smaller surrogate model offers cheap and reliable indications of the benefits achieved by adding these tasks.

Proxy	Target	Pearson	p-value
50 epochs	150 epochs	0.55	4.23e-3
Resnet18	Resnet50	0.94	1.42e-12
Clean performance	Robustness	0.98	1.91e-17

Table 5.3: Pearson correlation between the real adversarial vulnerabilities and proxy values from three different methods.

5.7 Threats to validity

Our study focused only on one dataset to evaluate the factors behind the robustness of multi-task models, however, we proposed a theoretical evaluation that relies on the least number of hypotheses to cover all multi-task learning paradigms. We also studied a large number of tasks and a diverse set of architectures, with different families and different sizes.

To mitigate the risk of coding errors, we base our implementation and experiments on existing software. The training of the models rely on the models and architectures shared by [ZSS+18]. The source code to evaluate the adversarial vulnerability is the one proposed by [SOB+19].

5.8 Conclusion

We have presented what is to date the largest evaluation of the vulnerability of multi-task models to adversarial attacks. Our study does not entirely reject the

benefits of adding auxiliary tasks to improve robustness, but rather tones down the generality of this proposition.

We evaluate different settings of multi-task learning, with a large combination of tasks, architectures, attack strengths and norms and show that in multiple settings, multi-task learning fails to protect against gradient attacks.

We also demonstrate that weight optimization can significantly improve the robustness of multi-task models, however, falls short to protecting against adaptive attacks for some tasks. In particular, we propose a new adaptive attack, WGD, that balances the gradient of the tasks and overcomes the gradient masking defense of multi-task learning.

Taking the perspective of the defender, we show that one can identify the most robust combinations of tasks efficiently by working on cheap surrogates.

Overall, our research contributes to guiding practitioners in the development of robust multi-task models and paves the way for methods to improve together the clean performance and the robustness of multi-task models.

6

ATTA: Improving Adversarial Training with Task Augmentation.

While leveraging additional training data is well established to improve adversarial robustness, it incurs the unavoidable cost of data collection and the heavy computation to train models. To mitigate the costs, we propose Adversarial Training with Task Augmentation (ATTA), a new adversarial training technique that exploits auxiliary tasks under a limited set of training data. ATTA is particularly useful in cases like medical imaging where some pathologies are available in limited samples. Our approach extends single-task models into multitasking models during the min-max optimization of adversarial training. ATTA leverages two types of auxiliary tasks: self-supervised tasks, where the labels are generated automatically, and domain-knowledge tasks, where human experts provide additional labels. Experimentally, under limited data, ATTA increases the robust accuracy on CIFAR-10 up to four times (from 11% to 42% robust accuracy) and the robust AUC of scarce pathologies in the CheXpert medical imaging dataset from 50% to 83%. On the full CIFAR-10 dataset, ATTA with an adaptive weighting strategy, further enhances the adversarial robustness achieved by common data augmentation techniques up to 48% robust accuracy.

Contents

6.1 Introduction

Despite their impressive performance, Deep Neural Networks (DNNs) are sensitive to small, imperceptible perturbations in the input. The resulting *adversarial inputs* raise multiple questions about the robustness of such systems, especially in safety-critical domains such as autonomous driving [CXC+19] and medical imaging [MNG+21].

Adversarial training (AT) [MMS+17c] is the de facto standard for building robust models. In its simplest form, AT trains the model using both original inputs and adversarial inputs. This process requires a large set of original labeled data to build robust models [MMS+17c]. It can achieve up to 66% of robust accuracy [CAS+20] by increasing the size of the models and the set of training data (with unlabeled data [NMK+19], augmented data [RGC+21], or artificial data from generative models [GRW+21]).

However, because it requires large amounts of data, AT is not effective for tasks where data is limited. Such scenarios occur, e.g., in the medical imaging, where some pathology labels remain scarce in the training sets. Even worse, AT with data augmentation and larger models has diminishing returns, and the adversarial robustness achieved with these approaches has already reached a computational plateau [SST+18; GRW+21].

In this paper, we propose a novel improvement to AT: *task augmentation*. Our approach, **Adversarial Training with Task Augmentation (ATTA)**, adds carefully-chosen auxiliary tasks to an original single-task model during the min-max optimization of AT. Our key hypothesis is that AT on multiple tasks regularizes the learning and introduces an inductive bias. This regularization, in turn, reduces the model's sensitivity to adversarial perturbations. Our approach shines in the scenario where available training data is scarce and improves the robustness achieved with data augmentation techniques. Ultimately, the combination of ATTA with state-of-the-art AT improvements paves the way for breaking the robustness ceiling that these past improvements alone have reached.

We evaluate ATTA on the CIFAR-10 and the CheXpert X-ray classification dataset. Each radiograph of CheXpert is multi-labeled for a dozen pathologies and contains meta-data about the patients' age, gender, or race. Our results reveal that:

- ATTA improves robustness under a limited training data budget. On CIFAR-10 classification, robust accuracy improves up to four times (from 11% to 42%). On the CheXpert medical image dataset, ATTA improves the robustness of scarce pathologies from 50% to 83%.
- On the full CIFAR-10 dataset, ATTA with adaptive weighting outperforms all data augmentation strategies. Furthermore, we can combine ATTA with different data augmentation strategies to further enhance robust accuracy

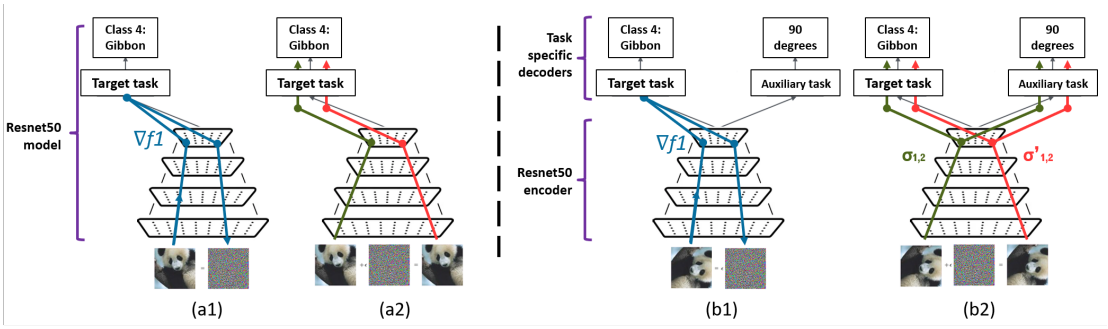


Figure 6.1: Comparison of single-task adversarial training (a) and our proposed approach ATTA (b). ATTA preserves the original target task and adds an auxiliary task where abundant labels are available: For instance, a self-supervised task like rotation angle prediction. In (a1) and (b1), we generate the adversarial example using only the loss of the target task (blue line). We update the models' weights with backpropagation in (a2) and (b2). We compute the model's weights update with ATTA (b2) using a weighted combination $(\sigma_{1,2}, \sigma'_{1,2})$ of the loss of the different tasks over the clean examples (green line) and the adversarial examples (red line).

from 22% to 48%.

- ATTA significantly differs in effectiveness depending on the chosen auxiliary tasks. The *gradient curvature measure* that we propose is the best auxiliary task selection metric over all our experiments.

6.2 Adversarial Training with Task Augmentation

In this section, we will first provide a motivating example where AT with an auxiliary task is extremely valuable (Section 6.2.1). We will then introduce our framework, ATTA, in Section 6.2.2 and its adaptive variant W-ATTA in Section 6.2.3. Finally, we provide in Section 6.2.4 three metrics that we hypothesize can discriminate the auxiliary tasks and drive their selection.

6.2.1 Motivation: Robust diagnosis of scarce pathologies

Medical imaging falls in the high dimensional, low sample size settings [AM10], where the amount of labeled data is limited, but the images are of large dimensions and annotated with dozens of labels and meta-data. It is, however, one of the fields where robustness to small perturbations is critical and tightly related to the generalization properties of the models [KHS22].

Data augmentations, especially with generative models, are at the early stages

in medical imaging [GRL+19]. They still require collecting a large and diverse set of images from patients of different demographics and acquisition protocols to train the generative models. However, this process remains costly, labor-intensive, and under strict regulations.

Furthermore, collecting diverse sets of images pre-treatment can be arduous for some pathologies, for instance, for pathologies that require immediate treatment. The practitioners can only collect and analyze images a-posteriori for more detailed labels (pathologies and stages) or enrich the inputs with meta-data like age, gender, race, co-morbidities, etc.

In the NIH dataset [WPL+17b], ML practitioners have access to 112,200 images, but only 82 images have been labeled with an Edema; the remaining are either negative or non-labeled. In the CheXpert dataset [IRa19] (224k images), only 12,691 have been labeled positively for Atelectasis; the remaining are negative samples, uncertain labels, or non labeled.

Our approach, ATTA, can leverage AT over multiple labels and tasks to balance the lack of samples for the target task (here, scarce pathologies). We hypothesize that our approach can significantly improve the robustness of scarce pathologies without additional training data.

6.2.2 The proposed approach: ATTA

ATTA transforms any single-task model into a multi-task model before AT. We connect additional decoders to the penultimate layer of the existing model. The architecture of each decoder is selected for one auxiliary task specifically. For example, we use a single dense layer as a decoder for classification tasks and a U-net[RFB15] decoder for segmentation tasks. In Figure 6.1, we extend an ImageNet classification model into a multi-task model that learns both the class (target task) and the orientation (auxiliary task) of the image. The auxiliary task here is "rotation angle prediction", a self-supervised task where we can generate the labels on the fly by rotating the original image.

We consider two types of task augmentation. In *self-supervised* task augmentation, the image is pre-processed with some image transformation like jigsaw scrambling [NF16] or image rotation [GSK18]. The auxiliary task predicts the applied image transformation (e.g., the permutation matrix for the jigsaw task, the rotation angle for the rotation task). In *domain-knowledge* tasks, a human oracle provides additional labels. In the medical imaging case, these additional labels may include, e.g., other pathologies and stages or patient data like gender and age.

In ATTA, the classical AT (Eq. (3.2)) is extended into the following min-max optimization problem:

$$\min_{\theta} \sum_{i} \max_{\delta \in \Delta} \sum_{j=1}^{M} (\mathscr{L}_j(x_i, y_{i,j}) + \mathscr{L}_j(x_i + \delta, y_{i,j})), \tag{6.1}$$

where $y_{i,j}$ is the label of the input example i for the task j.

In its simplest form, ATTA requires one target task and one auxiliary task. It then boils down to the following optimization:

$$\min_{\theta} \sum_{i} \max_{\delta \in \Delta} \Big[\mathscr{L}_j(x_i, y_{i,target}) + \mathscr{L}_j(x_i + \delta, y_{i,target}) + \mathscr{L}_j(x_i, y_{i,\text{aux}}) + \mathscr{L}_j(x_i + \delta, y_{i,\text{aux}}) \Big]. \tag{6.2}$$

We later use this form throughout our evaluation of ATTA.

6.2.3 The adaptive approach: W-ATTA

The ATTA optimization proposed in Eq. (6.1) faces conflicting gradients between the clean and the adversarial losses and possibly between the target and auxiliary tasks. Weighting strategies for MTL [LLK+21; YKG+20; WTF+20] all assume that the tasks' gradients are misaligned and not totally opposed. The case of ATTA is more complex because there is no guarantee that this assumption holds across the AT optimization. Therefore, instead of achieving the minimization of the whole loss, we seek to reach a Pareto-stationary point where we cannot improve the term of one task without degrading the term of another task.

To solve this multi-objective optimization problem, we get inspiration from the Multi-Gradient Descent Algorithm (MGDA) proposed by [Des12]. MGDA generalizes gradient descent to multi-objective settings by identifying a descent direction common to all objectives. It formally guarantees convergence to a Pareto-stationary point.

In addition, we hypothesize that variable task weights can improve the effectiveness of ATTA. This hypothesis is supported by the fact that, in MTL, task weights can drastically impact clean performance [ZSS+18; VGV+21]. Therefore, we extend our original ATTA to introduce weights over the tasks:

$$\min_{\theta} \sum_{i} \max_{\delta \in \Delta} \sum_{j=1}^{M} (w_j^{(clean)} \mathscr{L}_j(x_i, y_{i,j}) + w_j^{(adv)} \mathscr{L}_j(x_i + \delta, y_{i,j})), \tag{6.3}$$

where $w_j^{(clean)}$ and $w_j^{(adv)}$ are positive weights that control the relative contribution of the clean and adversarial loss (respectively) of task j to the function to optimize. We name this approach *Weighted ATTA* (W-ATTA).

We adapt MGDA to find a Pareto-stationary point for Eq(6.3). The clean and adversarial loss terms of all tasks are the functions to optimize and the optimization variables are the model parameters and the set of weights $\{w_j^{(clean)}, w_j^{(adv)}\}_{j=1}^{M}$. In doing so, we optimize a set of potentially conflicting gradients of different magnitudes. We also included in our method the improvements to the original MGDA that [SK18] proposed to increase computational efficiency.

We present in more detail the W-ATTA optimization in Algorithm 2.

Given: a single task model $\mathscr{M}$, a data loader $\mathscr{L}$ returning a training batch example x, and $\bar{y} = (y_1, ..., y_s, ...y_m)$ its corresponding ground-truth labels for each task, with y_1 the target task, $y_{1<i\leq s}$ the auxiliary self-supervised tasks and $y_{s<i\leq m}$ the auxiliary domain-knowledge tasks;

Given: an input processing f_t for each auxiliary self-supervised t task with label $y_{1<t\leq s}$.

Given: a weight optimizer *opt*; a list of task-specific decoders functions $\mathscr{D} = \{D_1, ..., D_M\}$, each decoder uses it specific loss function (L1, CE, MSE...);

Given: a PGD adversarial attack with a step size ϵ_{step}; a maximum perturbation size ϵ; one random start and a total number of attack iterations S;

Step 1: Create multiple branches at the penultimate layer of $\mathscr{M}$. Each branch consists of the decoder D_i for of the auxiliary task t_i / $i > 1$.

Step 2: For each epoch and for each batch of the data loader $\mathscr{L}$ Do

1. For each self-supervised task $t_{1<i\leq s}$, successively pre-process the batch examples x with the appropriate input processing function: $x \leftarrow \bigcirc_{t=2}^{s} \ f_t(x)$
2. Generate $\hat{x}$, the adversarial examples of x: $\hat{x} \leftarrow \text{PGD}(x, y_1, \epsilon_{step}, \epsilon, S)$.
3. Compute the losses $l_{i,x}$ and $l_{i,\hat{x}}$ of x and $\hat{x}$ respectively for each task t_i with label y_i; $l \leftarrow [l_{1,x}, l_{1,\hat{x}}, ..., l_{M,x}, l_{M,\hat{x}},]$.
4. Compute $\nabla l_{i,x}$ the normalized gradient of each loss in l
5. Apply MGDA to find the minimum norm element in the convex hull given the list of losses.
6. Back-propagate the weighted gradients and update the model weights with optimizer *opt*.

Step 3: Disable the auxiliary branches added at step 1.

Algorithm 2: Meta-Algorithm of Weighted Adversarial Training with Task Augmentation

6.2.4 Selection of auxiliary tasks

In standard MTL, the relative properties of the task gradients – such as orientation angle, magnitude similarity, and curvature – have an impact on learning speed and on the achieved clean accuracy [VGV+21]. Therefore, task weighting approaches like PCG [YKG+20] rely on these properties to optimize classical training.

We hypothesize that these same properties can help select effective auxiliary tasks for W-ATTA. Our intuition is that AT changes these properties and that the properties are (negatively) correlated to robustness. Therefore, the best tasks to use as auxiliary maximize (or minimize) these properties. We propose to use the following metrics to drive task selection and empirically check their correlation to robustness in our evaluation.

Definition 5. *Let ϕ_{ij} be the angle between two tasks' gradients $\mathbf{g}_i$ and $\mathbf{g}_j$. We define the gradients as* ***conflicting*** *when* $\cos\phi_{ij} < 0$.

Definition 6. *The* ***gradient magnitude similarity*** *between two gradients $\mathbf{g}_i$ and $\mathbf{g}_j$ is* $\Phi(\mathbf{g}_i, \mathbf{g}_j) = \frac{2\|\mathbf{g}_i\|_2\|\mathbf{g}_j\|_2}{\|\mathbf{g}_i\|_2^2+\|\mathbf{g}_j\|_2^2}$.

When the magnitude of two gradients is the same, this value equals 1. As the gradient magnitude difference increases, the similarity goes towards zero.

Definition 7. *The* ***multi-task curvature bounding measure*** *between two gradients $\mathbf{g}_i$ and $\mathbf{g}_j$ is* $\xi(\mathbf{g}_i, \mathbf{g}_j) = (1 - \cos^2\phi_{12})\frac{\|\mathbf{g}_i-\mathbf{g}_j|_2^2}{\|\mathbf{g}_i+\mathbf{g}_j\|_2^2}$.

The multi-task curvature bounding measure combines information about both the orientation of the gradients of the tasks and the relative amplitude of the gradients [YKG+20]. We show in Section 6.6 that this metric has strong negative correlations with model robustness

6.3 Experimental setup

We provide further details of the experimental settings in Appendix A.

Dataset. Our evaluation focuses on CIFAR-10 dataset [KH+09], 32x32 color image dataset. We evaluate two scenarios: A full 50.000 image adversarial training scenario and an adversarial training scenario using a subset of 10%. A study with 25%, and 50% of the original training data is in Appendix B.

We extend the evaluation to a large public chest X-ray dataset: *CheXpert* [IRa19]. It consists of 512x512 grayscale image radiography collected from one hospital. We restrict our evaluation in the main paper to predicting the Edema and the Atelectasis disease as target tasks in separate models. We provide the results for other combinations of pathologies in Appendix B.

Architecture. We use an encoder-decoder image classification architecture, with ResNet-50v2 [HZR+16] as encoders for the main study. We provide in Appendix C complementary studies with WideResnet28-10 and WideResnet70-16 [ZK16] encoders.

Task augmentations. For both CIFAR-10 and CheXpert datasets, we evaluate two self-supervised tasks: **Jigsaw**, where we split the images into 16 chunks and scramble them according to a permutation matrix. The permutation matrix represents the labels of the Jigsaw prediction task. In the **Rotation** auxiliary task, we rotate the images by 0, 90, 180, or degrees, and the 4 rotation angles are the labels learned by the Rotation prediction task.

To evaluate domain knowledge tasks, we generate new labels as follows. For CIFAR-10, we split the existing 10 classes into 2 macro classes: *Vehicles* or *Animals*. We refer to this task as **Macro**. For CheXpert, we add the binary classification of **Cardiomegaly** and **Pneumothorax** as auxiliary tasks. These auxiliary pathologies often co-occur with Edema and Atelectasis. We also extract the age and gender meta-data related to the patients and use them to build auxiliary tasks. Learning the **Age** is a regression task, while learning the **Gender** is a 3-class classification task.

Adversarial Training. Both Natural and adversarial training is combined with common data augmentations (rotation, cropping, scaling), using SGD with lr=0.1, a cosine annealing, and early stopping. We train CIFAR-10 models for 400 epochs and CheXpert models for 200 epochs. We perform AT following Madry's approach [MMS+17c] with a 10 steps PGD attack and $\epsilon = 8/255$ size budgets, and we only target the main task to craft the adversarial examples.

6.4 RQ1: ATTA can significantly improve adversarial robustness under limited data

ATTA increases up to 4 times the robustness of CIFAR-10 models with partial data. We restrict the adversarial training of the models to 10% of the full CIFAR-10 training dataset and compare the performance of the single-task adversarially trained models and the multitasking models trained with ATTA. We provide in Figure 6.2a the average across three runs and seeds of each model. We evaluate the robust accuracy against PGD-10 (in blue) and PGD-4 (in orange). Appendix B presents the detailed results for 10%, 25%, and 50% of the training data.

ATTA with self-supervised tasks and ATTA with domain-knowledge tasks both outperform single-task model adversarial training. In particular, the **Macro** task augmentation boosts the robust accuracy from 8.37% to 22.42% against PGD-10 and from 11.81% to 42.68% against PGD-4.

When comparing the robustness achieved by ATTA under PGD-4 and PGD-10 attacks, our results show that some tasks are more robust against stronger attacks. AT with **Jigsaw** task shows only a 0.15% drop when we increase the strength of the attack from 4 to 10 iteration attacks. Subsequently, the proper selection of the auxiliary task is critical in limited data scenarios.

ATTA increases up to 41% the robustness of medical diagnosis for scarce pathologies. Figure 6.2b shows the clean and robust AUC of the single-task baseline models (circle marker), and the task-augmented models.

For Atelectasis (blue), **age** task augmentation leads to lower results than the baseline. Meanwhile, all remaining task augmentations outperform the baseline both on the clean and robust AUC. The gender augmentation in particular increases the robust AUC of Atelectasis from 58.75% to 83.34%.

For Edema (orange), Task augmentation with Jigsaw leads to the best clean and robust AUC increase. The robust AUC jumps from 55.68% to 70.47% compared to single-task adversarial training.

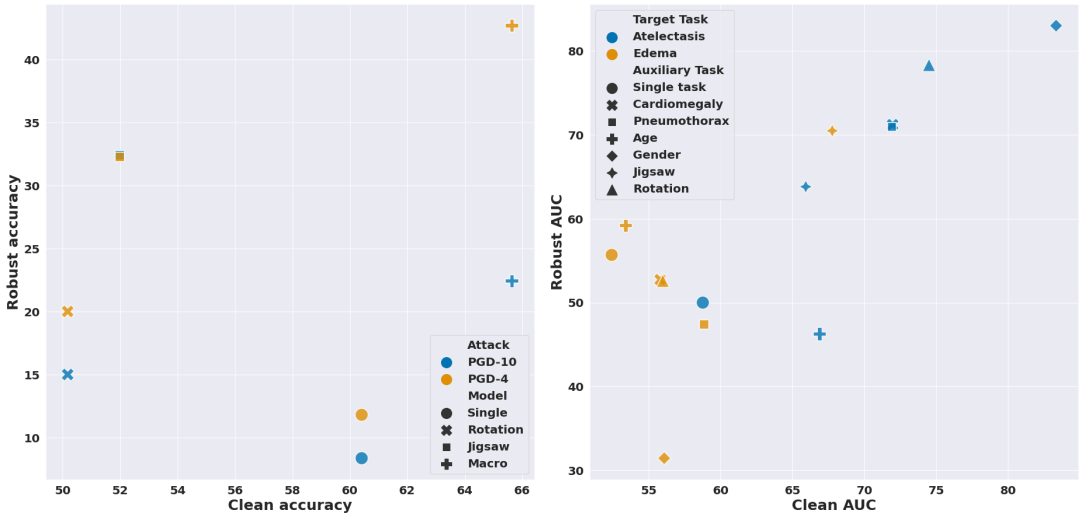
When we compare the improvements obtained on Edema, Atelectasis, and CIFAR-10, our evaluation shows that no one winning auxiliary task outperforms the others for all target tasks. The most effective task augmentation depends on the target task we aim to improve. Its identification motivates our exploration of the relevant metrics that can drive the selection of the auxiliary task.

6.5 RQ2: ATTA is complementary with data augmentation strategies in the full training data setting

ATTA improves the robustness of models over data augmentation strategies. We compare in Table 6.1 the clean and robust accuracy of AT with data augmentation to AT with Weighted Task Augmentation. We evaluate below W-ATTA (i.e., ATTA optimized with MGDA weighting) and provide in Appendix C a complementary study of other weighting strategies that remain less effective than MGDA.

Compared to the single-task model, ATTA with a **Macro** auxiliary task shows an improvement of 9.29% in robust accuracy and only a drop of 0.79% in clean accuracy. In addition, **Macro** auxiliary task outperforms all data augmentation strategies for robust accuracy but achieves, in comparison, a lower clean accuracy.

W-ATTA and data augmentation strategies can be combined to improve clean and robust performances. When we combine W-ATTA with various data augmentation strategies, the robust accuracy of the models is higher than the robustness of the models with data augmentation alone in seven of the nine



(a) Clean and robust accuracy for CIFAR-10

(b) Clean and robust AUC for scarce pathologies

Figure 6.2: Comparison of different Task Augmentation strategies with single-task models using Adversarial Training; Clean and robust performance of ATTA vs Single task adversarial training. (a) shows the accuracy of CIFAR-10 models adversarially trained with a 10% subset of data.
(b) shows the AUC of models trained to diagnose Atelectasis and Edema pathologies.

cases (in blue, Table 6.1). The two exceptions are CutMix when combined with **Jigsaw** or **Rotation**. However, in most cases, W-ATTA alone provides better robustness than when combined with data augmentation strategies. The main benefit of combining W-ATTA with data augmentation is that their combination significantly improves clean accuracy, for example, from 56.51% to 87.86% using the **Rotation** auxiliary task. For all the nine combinations of W-ATTA and data augmentations, the clean accuracy is improved over W-ATTA alone (in underline, Table 6.1).

6.6 RQ3: Selection of auxiliary task

Our previous evaluations showed that the selection of the tasks could significantly impact the final robustness of the models. It raises the question about the metrics that can help select the best auxiliary task for the ATTA.

We evaluate in Fig. 6.3 the Pearson correlation coefficient between the robust accuracy and each of the three evaluated metrics: We evaluate in the top figure models we adversarially trained in the previous sections. Tn the bottom figures, we evaluate the same models but trained with standard training. The vertical

Table 6.1: Combination of our approach (W-ATTA) with data augmentation techniques. The blue cells indicate the combinations that outperform data augmentation techniques alone, the underlined cells are the combinations that outperform task augmentation alone and, in bold the best performances.

Task	Robust accuracy (%)				Clean accuracy (%)			
Augmentation	None	Jigsaw	Macro	Rotation	None	Jigsaw	Macro	Rotation
None	39.09	32.95	**48.38**	36.13	74.49	43.99	73.70	56.51
Cutmix	38.95	23.86	41.09	20.19	87.31	60.53	87.52	**87.86**
Unlabelled	21.98	26.33	33.88	35.29	87.31	49.32	84.57	71.08
Pre-train	27.30	35.47	32.64	33.78	86.64	87.56	86.69	87.85

axis consists of the measured metrics, and the horizontal axis refers to the robust accuracy.

For adversarially trained models (top), both the **Gradient multi-task curvature bounding measure** (left) and the **Gradient cosine angle**(right) are strongly negatively correlated with the adversarial robustness, with respectively a correlation coefficient r of -0.86 and -0.87. However, for models trained with standard training, only the **Gradient multi-task curvature bounding measure** is moderately correlated (r= -0.45)to the robustness of the models, with a p-value of 0.09.

Our results confirm that the **gradient curvature measure** is the best candidate to drive the training towards more robust models, especially with adversarial training.

6.6.1 Generalizing to other threat models, architectures, and datasets

Table 6.2: Robust accuracy (%) of different models adversarially trained with ATTA, with 3 different task augmentations, compared to their counterpart single task adversarially trained models. In bold the cases where ATTA outperforms single-task AT.

Dataset	Scenario	Auxiliary task			
		None	*Jigsaw*	*Macro*	*Rotation*
CIFAR-10	AutoAttack	27.01	**29.63**	**32.54**	13.82
	PGD-10 surrogate attack	1.53	**13.44**	**10.8**	**15.45**
	WideResnet28-10	42.52	32.75	**46.6**	41.06
STL-10	PGD-10 whitebox attack	18.7	**36.02**	**19.72**	**34.65**

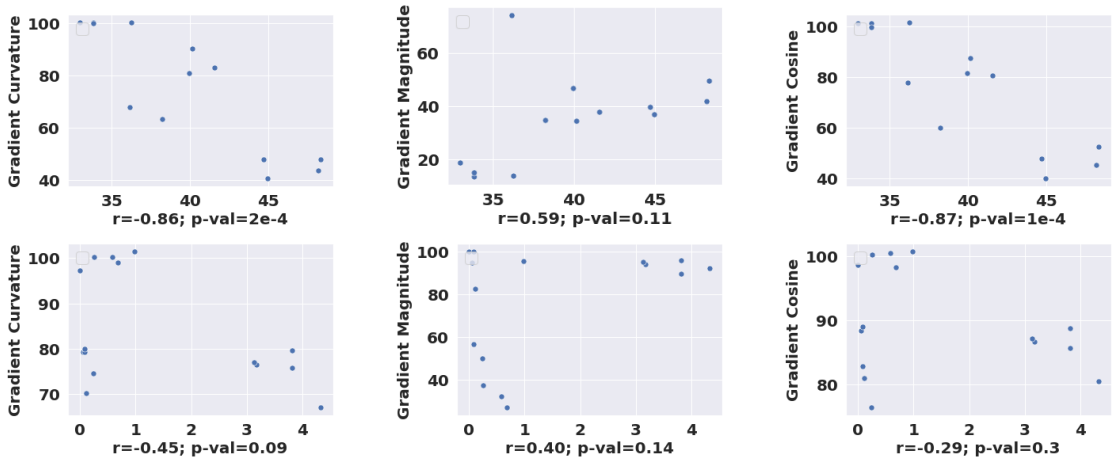


Figure 6.3: Evolution of the robust accuracy (Y-axis) with each of our three metrics (Y-axis). Top: Models with adversarial training, bottom: Models with standard training. Left: Gradient multi-task curvature bounding measure, middle: Gradient magnitude similarity, right: Gradient cosine angle. Below each scatter plot is the Pearson correlation coefficient r and its p-value between the robust accuracy and the studied metric.

We summarize our generalization studies in Table 10.14, and provide extensive figures in Appendix C.

Generalization to adaptive attacks. To assess if Adversarial Training with Task Augmentation is not a gradient-obfuscation defense, we evaluate our defended models against AutoAttack [CH20b]. This attack combines gradient-based and blackbox attacks in targeted and untargeted threat models. The results in Table 10.14 show that ATTA with **Jigsaw** or **Macro** auxiliary tasks provide better robustness to AutoAttack than AT with the target task alone. Thus, we confirm that stronger attacks do not easily overcome the robustness provided by ATTA.

Generalization to surrogate attacks. We evaluate in Table 10.14 the threat model where the attacker has access to the full training set but has no knowledge of the auxiliary tasks leveraged by ATTA. Models trained with ATTA have slightly different decision boundaries from models with common AT. The success rate of surrogate attacks drops from 98.47% (i.e., 1.53% robust accuracy) to 84.55% when we train the target task with **Rotation** based ATTA.

Generalization to WideResnet architectures given the same computation budget as Resnet50. We train for 150 epochs WideResnet28-10 models with ATTA and compare their robust accuracy to a single-task WideResnet28-10 model

with AT. **Macro** increases the single-task model's robust accuracy from 42.52% to 46.6%.

Generalization to STL-10 dataset. STL-10 [CNL11] dataset is a subset of ImageNet resized to 96x96px, and restricted to the same classes as CIFAR-10. Each class has 500 training images (10% of CIFAR-10). Therefore, STL-10 fits our study's scope of robust learning with limited training data. Table 10.14 shows that ATTA improves the robustness of that target task with all the proposed augmentations. In particular, the robust accuracy jumps from 18.7% to 36.02% using the **Jigsaw** augmentation.

6.7 Conclusion

In this paper, we demonstrated that augmenting single-task models with self-supervised and domain-knowledge auxiliary tasks significantly improves the robust accuracy of classification models with limited training data. We proposed a novel adversarial training approach, Weighted Adversarial Training with Task Augmentation that solves the min-max optimization of adversarial training through the prism of Pareto multi-objective learning. Our approach complements existing data augmentation techniques for robust learning and improves adversarially trained models' clean and robust accuracy. We expect that combining data augmentation and task augmentation will break ground in bettering the adversarial robustness beyond the ceiling of current adversarial training approaches.

7

Towards Generalizable Machine Learning for Chest X-ray Diagnosis with Multi-task learning.

Clinicians use chest radiography (CXR) to diagnose common pathologies. Automated classification of these diseases can expedite analysis workflow, scale to growing numbers of patients and reduce healthcare costs. While research has produced classification models that perform well on a given dataset, the same models lack generalization on different datasets. This reduces confidence that these models can be reliably deployed across various clinical settings. We propose ***Auxiliary Pathology Learning (APL)****, an approach based on multi-task learning to improve model generalization. We demonstrate that learning a (main) pathology together with an auxiliary pathology can significantly impact generalization performance (between -10% and +15% AUC-ROC). A careful choice of auxiliary pathology even yields competitive performance with state-of-the-art models that rely on fine-tuning or ensemble learning, using between 6% and 34% of the training data that these models required. Finally, we suggest two cheap surrogates to select the auxiliary pathologies that lead to the highest generalization performance.*

Contents

7.1 Introduction

The recent success of ML techniques for image analysis has extended to medical imaging, notably to assist radiologists when diagnosing pathologies. Nevertheless, medical data solutions present two specific challenges that have to be solved before these solutions reliably run in clinical settings. First, medical images differ from classical ML datasets as they present higher dimensionality (3D scans, for instance) and increased uncertainty in the labeling process ([BPA17]). Second, medical data tend to be not only scarce but also partially represented ([VC21]).

Furthermore, deploying an ML technique in a target population different from its training population may result in data for training and evaluation with different distributions, thus, breaking the independent and identically assumption [DH20]. These challenges affect model *generalizability*. Generalizability is essential in the medical domain because practitioners need models that provide stable predictions and can efficiently adapt to new clinical settings (e.g., different hospitals, different countries, different populations, and so forth and so) at an affordable computation cost.

An ML model is said to be **generalizable from a source to a target population** when its performance metrics on the target population do not significantly drop compared to its performance on the train and test populations. In such settings, existing models do not always generalize ([CHB+20]) – i.e., a model trained to predict pathologies on the source population has poor performance on the target population. As a result, the lack of generalization of existing models hampers their safe and accurate translation into clinical trials ([BBa20]).

Experimentally, a common way to assess model generalizability is to train a model on the training set of a *source* dataset (which represents the original population that the model learns from), then to evaluate its performance on the test set of a target dataset ([YPC+19]). We expect the target dataset to represent the population on which we will deploy the model.

Because the target dataset's distribution may differ from the source, it can be used as a proxy to simulate different clinical contexts. However, most of the ML for CXR literature focuses on standardized benchmarks that cover a single dataset, *e.g.* by winning the CheXpert competition([IRa19]) [1].

Our research focuses on the study of model generalization in medical image analysis, particularly chest radiographs (CXRs), and brings three contributions. Our first contribution is to demonstrate that learning multiple pathologies together in a multi-task model can significantly impact the generalization performance of each pathology while imperceptibly impacting test performance.

While multi-task learning has been previously proposed to improve test per-

[1] `https://stanfordmlgroup.github.io/competitions/chexpert/`

formances of models. To the best of our knowledge, we are the first to connect multi-task learning and generalization for CXR classification. Our work is the first to demonstrate that multi-task learning can imperceptibly impact test performance, which may cause practitioners to disregard it, but significantly impact generalization performance.

Through our extensive experiments, we aim to demonstrate that some pathologies can be consistently learned with improved generalization performance when learned with multi-task learning. Our second contribution is to propose *Auxiliary Pathology Learning*, a method to improve medical model generalization via a multi-task model. Our approach learns a main pathology of interest with the support of a carefully selected auxiliary pathology.

Finally, we suggest a set of cheaper surrogates to select an auxiliary pathology. First, we demonstrate a strong correlation between the generalization performance of a model trained with a main and auxiliary pathology and the test performance when fine-tuning the model from the auxiliary pathology to the main pathology. Next, we identify patterns in the layer similarity of models specific for pairwise combinations that are robust, and patterns specific to pathology combinations that are not. Therefore, we can use these approaches to drive the selection of the best auxiliary pathology (for a given main pathology) using only the source dataset.

This paper is structured as follows: First, we present our approach *Auxiliary Pathology Learning (APL)* in section 7.2.1, and its associated algorithm (Section 7.2.3). Then we introduce our experimental protocol: the datasets (section 7.2.4), the models (section 7.2.5) and the metrics of our study (section 7.2.6). We also present in section 7.2.7 our surrogate techniques to identify the most relevant auxiliary pathologies.

Next we present the results of our extensive study of the generalization of models using *Auxiliary Pathology Learning* in section 7.3 and a comparison with SoTA models in section 7.4. Finally, in section 7.5, we evaluate our surrogate techniques and study their correlation to the robust performance of the models.

We discuss our results in section 7.6, and showcase how our approach can help practitioners efficiently build CXR classification models, even for pathologies with scarce training data.

7.2 Material and Methods

7.2.1 Overview of our approach: Auxiliary Pathology Learning

We propose to envision multilabel CXR classification as a multi-task binary problem where the main task is our target pathology, and the auxiliary task is the pathology used for augmentation.

The motivation behind building multi-task models is threefold: (1) multi-task models allow combining within the same model tasks of different nature: Image classification, image segmentation (pathology segmentation, for example), and Regression (Age prediction). (2) multi-task models allow a fine-tuned control of the weight of each pathology through the training. Because some pathologies are easier to learn, the loss computed over the pathologies may not always be well balanced. (3) Contrary to ensemble learning, the shared encoder across the pathologies learns a common representation and leverages common knowledge to improve the generalization of the models.

Auxiliary Pathology Learning consists of a multi-task model where the encoder uses one of the standard architectures in CXR classification (Resnet, Densenet), and the decoder contains two heads. The first head comprises one single dense layer, and its activation outputs binary logits. The second head consists of a single dense layer for binary classification tasks (e.g., predicting the presence of the auxiliary pathology), and its activation outputs sigmoid logits.

The common encoder extracts the features most relevant to the set of pathologies. Each pathology has a dedicated decoder that learns pathology-specific weights and outputs the final probability of this pathology. We denote by **main pathology** the pathology we aim to evaluate on the target population and **auxiliary pathology** the secondary pathology we include as an auxiliary task.

When the labels of both tasks are available, Auxiliary Pathology Learning can adjust the weights of each task according to a given heuristic (for example, using MGDA).

7.2.2 Problem Definition

Formally, we consider CXR image x. We denote by $\bar{y}$ the corresponding ground-truth label, defined as $\bar{y} = (y_1, ..., y_i, y_M)$ where y_i is the corresponding ground truth for task i (*i.e.*, pathology i) for an input x. For a given population, x and $\bar{y}$ are drawn from some joint distribution $p(x, \bar{y})$.

Let $\mathscr{M}$ be a multi-task model with tasks $\mathscr{T} = \{t_1, ..., t_M\}$. $\mathscr{M}$ is trained to estimate $p(\bar{y} \mid x)$ but may not generalize well when the joint distribution changes. For instance, when the population of patients changes, or the collection protocol (hospitals, machines, ...) varies. We hypothesize that $p(\bar{y} \mid x)$ is not consistent across datasets, and we propose to consider a more fine-grained problem: Training a multi-task model $\mathscr{M}$ to learn to estimate $p(y_j \mid x, y_k)$ where $t_j, t_k \in \mathscr{T}^2$ two pathologies that are learned together by our model $\mathscr{M}$. Using an auxiliary pathology to learn a main one could mitigate the covariate shift that [CHB+20] previously uncovered.

The total loss $\mathscr{L}$ of our multi-task model is a weighted sum of the individual

losses $\mathscr{L}_i$ of each task:

$$\mathscr{L}(x, \bar{y}) = \sum_{i=1}^{M} w_i \mathscr{L}_i(x, y_i)$$

where $\{w_1 \dots w_M\}$ are the weights of the tasks, either set manually or optimized during training [SK18]. We focus our study on the evaluation with equal weights, as task weighting strategies have been extensively explored in the literature [VGV+21] and showed no clear advantage over equal weighting strategy.

7.2.3 Algorithm

We present in 3 the pseudo-algorithm of our approach, starting from a single task CXR classification model.

Given: a single pathology model $\mathscr{M}$, a data loader $\mathscr{L}$ returning a training batch example x, and $\bar{y} = (y_1, ..., y_s, ...y_m)$ its corresponding ground-truth labels for each pathology, with y_1 the main pathology, $y_{1<i\leq M}$ the auxiliary pathologies;

Given: a weight optimizer *opt*; a list of task-specific decoders functions $\mathscr{D} = \{D_1, ..., D_M\}$, each decoder uses it specific loss function (L1, CE, MSE...);

Step 1: Create multiple branches at the penultimate layer of $\mathscr{M}$. Each branch consists of the decoder D_i for of the auxiliary pathology t_i / $i > 1$.

Step 2: For each epoch and for each batch of the data loader $\mathscr{L}$ Do

1. Compute the losses $l_{i,x}$ of x for each pathology t_i with confident label y_i; $\mathcal{L} \leftarrow [l_{1,x}, ..., l_{N,x},]$. The loss is not computed over non-confident labels.

2. Compute the optimal weights $\mathcal{W}_\mathcal{L}$ for each loss item in $\mathcal{L}$ using a weighting strategy (e.g., equal weights, MGDA, IMTL, ...).

3. Back-propagate the gradients with the optimal loss weights $\mathcal{W}_\mathcal{L}$ and update the model weights with optimizer *opt*.

Step 3: Disable the auxiliary pathologies added at step 1.

Algorithm 3: Pseudo-Algo of Auxiliary Pathology Learning

7.2.4 Data preparation

Datasets We run our evaluation on three large public chest X-ray datasets. NIH Chest X-ray14 [WPL+17a], denoted *NIH* in the following, is a dataset of 112k images partially labeled automatically with the NegBio labeler. *CheXpert*([IRa19]) (Chex), a set of 224k chest radiographs tagged with a custom automated labeler over the NLP analysis of radiology reports. *PadChest* [BPS+20] (PC) is a 160k

image dataset where, for 27% of them, the labels are extracted from radiographic reports manually annotated by physicians.

Similar to previous work on the CXR classification on these datasets [**chexpertHighRes**; CHB+20], we restrict our evaluation to the seven pathologies common to the three datasets: Atelectasis (ATE), cardiomegaly (CAR), consolidation (CON), edema (EDE), effusion (EFF), pneumonia (PNE), and pneumothorax (PTX). The datasets display a diverse pathology distribution, gender balance, and age distribution of the evaluated subjects, as detailed in Table 10.7.

Table 7.1: Characteristics of NIH, CheXpert, and PadChest datasets using in our trained models.

	NIH	CheXpert	PadChest
Number of radiographs	112,120	224,316	160,846
Number of patients	30,805	65,240	67,000
Age in years: mean (std)	46.9 (16.6)	60.7 (18.4)	58.5 (20)
Percentage of females (%)	43.5	40.6	49.7
Number of pathology labels	8	14	15

For each dataset, we restrict our study to the erect anteroposterior chest views (AP views) images. We use 80% of the images for the training and 20% for testing the performance as proposed in [CHB+20]. We present in Appendix A the demographics distributions and the pathologies of each dataset.

In the following, we call **source dataset** the dataset used to train the model, and **target dataset** the dataset used to evaluate the model's performance.

Data split We rely on TorchXrayVision library [CVB+21] to obtain the train and test sets. While CheXpert and PadChest naturally provide two distinct datasets, we use for NIH dataset 20% of the dataset as a test dataset and the remaining 80% for training. The training datasets of NIH, CheXpert, and PadChest are split between an actual train set and a validation set following a 70% / 30% split. All splits are generated randomly with a seed of 0.

Data augmentation Following [CHB+20], we use center cropping, a 15° random rotation, and a 15% random scaling and translation. We also randomly flip the images alongside the horizontal and vertical axis. Finally, we normalize the pixel values between 0 and 1.

7.2.5 Models and training

Our core evaluation required training 336 Deep Learning models across three datasets.

Encoders We focus our evaluation on ResNet50 encoders, as they are commonly used in the Chest X-ray literature [BNG+19; BAE+20]. We provide complementary results with Resnet34 and DenseNet121 encoders in appendix B.

Decoders We train single-pathology decoders (7 in total) and pairwise decoders (21) for each dataset. In addition, each single-pathology model is fine-tuned on each other pathology, with and without encoder freezing ($7 * 6 * 2 = 84$). Over the 3 datasets, we trained and evaluated a total of 336 models ($(7 + 21 + 84) \times 3$).

Training Models are trained for 250 epochs with a learning rate of 0.001 using the Adam optimizer following the protocol that [CHB+20] proposed. When multiple decoders are used, they are weighted equally, following common practice [ZSS+18]. The impact of task weighing on robustness is outside the scope of this work but has been covered by [GCP+21d]. For pathology decoders, the loss function is a binary cross-entropy.

For each dataset, we obtain the 84 fine-tuned models as follows: We start from the best performing model on an auxiliary pathology, then train it for an additional 10 epochs on the main pathology following the same protocol as above.

Pre-trained SOTA models SoTA models are provided and pre-trained by Cohen et al. [CHB+20]. They introduce different strategies to improve performance:

(1) **DenseNet** combines all pathologies and uses a densenet-121 architecture. Bressem et al. [BAE+20] have shown that Densenet121 is the best performing architecture for the classification of chest radiographs. It remains more expensive to train than Resnet-50 and is trained on all the pathologies of the CheXpert dataset (*i.e.*, 224,316 images)

(2) **EnsembleNet** uses an ensemble of 30 Densenet-121 models. This model is trained on five pathologies Atelectasis, Cardiomegaly, Consolidation, Edema, and Effusion. This model is trained on a total of 198.072 images.

(3) **MixtureNet** uses a ResNet-50 model but is trained on a mixture of 7 CXR datasets (cf. Appendix A) and all the pathologies, totaling a training set of 950,778 images.

7.2.6 Evaluation

Model performance We focus our core evaluation on the performance of the models using the area under the ROC curve metric (AUC) as it is the most standard metric for unbalanced binary classification([CHB+20; IRa19]). Additionally, we plot the receiver operation characteristic (ROC) curves for some models and provide in Appendix B extensive figures using the full ROC curves, including the FPR and the FNR values.

In the following, we call **test performance** the performance achieved by the model when the train and evaluation distributions are the same (i.e., coming from the same dataset NIH, CheXpert, or PadChest). We call **generalization**

performance the performance achieved by the model when the train and evaluation distributions are different (i.e., different datasets for train and evaluation among NIH, CheXpert, and PadChest).

For each of the seven pathologies, we train on CheXpert a model to predict a pair of pathologies and evaluate their prediction performance (*i.e.*, AUC) on CheXpert (source) and on the NIH dataset and PadChest dataset (target datasets). Due to space restrictions, we focus the study of the ROC curves on two pathologies: Edema and Pneumothorax, and provide the remaining figures in Appendix B.

Next, we compare the test and generalization performance of the SoTA models to the performance of our approach. We also compare the number of images each approach needs to achieve these performances.

Hidden representation analysis Neural network hidden representations are challenging to analyze because of their large neuron distribution and interactions across all layers. Kornblith et al. [KNL+19] proposed the **centered kernel alignment (CKA)** that provides a reliable quantitative measure of the similarity of neural network representation. It was later extended by [NRK21] for large and deep models.

Let $\mathbf{X} \in \mathbb{R}^{m \times n_1}$ and $\mathbf{Y} \in \mathbb{R}^{m \times n_2}$ the matrix representations of two layers, one with n_1 neurons and another n_2 neurons, to the same set of m examples. Each element of the $m \times m$ Gram matrices $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^\intercal$ and $\boldsymbol{L} = \boldsymbol{Y}\boldsymbol{Y}^\intercal$ reflects the similarities between a pair of examples according to the representations contained in $\boldsymbol{X}$ or $\boldsymbol{Y}$. Let $\boldsymbol{H} = \boldsymbol{I}_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\intercal$ be the centering matrix. Then $\boldsymbol{K}' = \boldsymbol{H}\boldsymbol{K}\boldsymbol{H}$ and $\boldsymbol{L}' = \boldsymbol{H}\boldsymbol{L}\boldsymbol{H}$ reflect the similarity matrices with their column and row means subtracted.

HSIC is defined as the similarity of these centered similarity matrices by reshaping them to vectors and taking the dot product between these vectors, $\text{HSIC}(\boldsymbol{K}, \boldsymbol{L}) = \frac{\text{vec}(\boldsymbol{K}')\text{vec}(\boldsymbol{L}')}{(m-1)^2}$. HSIC is invariant to orthogonal transformations of the representations and to permutation of neurons, but it is not invariant to scaling of the original representations. CKA further normalizes HSIC to produce a similarity index between 0 and 1 that is invariant to isotropic scaling,

$$\text{CKA}(\boldsymbol{K}, \boldsymbol{L}) = \frac{\text{HSIC}(\boldsymbol{K}, \boldsymbol{L})}{\sqrt{\text{HSIC}(\boldsymbol{K}, \boldsymbol{K})\text{HSIC}(\boldsymbol{L}, \boldsymbol{L})}}.$$

[KNL+19] showed that linear CKA between layers of architecturally identical networks trained from different initialization reliably identifies corresponding layers. We implement linear CKA following the mini-batch split proposed by [NRK21].

In the discussion, we analyze the interactions between pathologies through the prism of hidden representation similarity. Using the CKA metric over the test set on models trained with different combinations of pathologies, we evaluate the activation patterns of each of the 50 layers of our models. We obtain a 50x50 size

matrix, and each element (i,j) of the matrix is the layer similarity between layer i and layer j of the model. CKA is normalized and bounded between 0 and 1, with one meaning perfect similarity. The diagonals are always equal to 1, and high values at matrix element (i,j) mean that the two layers i and j show a high CKA similarity.

7.2.7 Surrogates

We evaluate four surrogate metrics to drive the selection of the auxiliary task that would lead to the best generalization performance. Our target metric is the change of generalization AUC of the model from the single pathology model to the auxiliary pathology model. I.e., how would the *generalization performance* of the model change when we add an auxiliary task.

We evaluate the correlation across all our models between the surrogate metric and our target metric for each surrogate metric. We report Spearman's Rank Correlation Coefficient R and its associated Probability Value (p).

1. Test performance (Test) Our first surrogate is the straightforward test performance metric. We evaluate if the combinations of pathologies that achieve the highest test AUC also achieve the highest generalization AUC. The metric we evaluate for this surrogate is the test AUC change of the main pathology between when the model is trained with the main and auxiliary pathology and when it is trained with only a single main pathology. I.e., how would the *test performance* of the model change if we add an auxiliary task.

2. Fine-tuning performance (Finetune) We hypothesize that the auxiliary pathologies that lead to the best generalization performance are also the ones that lead to the best test performance when used for pre-training. In our evaluation, we pre-train our models on each pathology and save the best performing checkpoint. Then we fine-tune the models for ten epochs on the main pathology. During these ten epochs, the weights of both the encoder and the decoder are updated. The metric we evaluate for this surrogate is the test AUC of the main pathology.

3. Fine-tuning performance with encoder freezing (Freeze) In this variant, we proceed as above, but we freeze the weights of the encoder. During the fine-tuning, only the weights of the decoder are updated. The metric we evaluate for this surrogate is the test AUC of the main pathology. The difference between this third approach and the second is that the former (encoder freezing) expects that the learned representation in the encoder between the auxiliary pathology and the main pathology are similar and do not require fine-tuning to achieve good performances. I.e., pathologies used in the pre-training and fine-tuning both exhibit the same low-level features.

Table 7.2: Comparison of AUC performance of our approach (APL), compared to the same models with single pathology or all pathologies trained on the CheXpert dataset and evaluated on CheXpert (left), NIH (middle), and PadChest (right).

	CheXpert → CheXpert			CheXpert → NIH			CheXpert → PadChest		
	APL	Single	All	APL	Single	All	APL	Single	All
Atelectasis	**89.59**	87.97	89.17	69.25	**70.77**	67.82	**69.55**	68.71	68.56
Cardiomegaly	**89.71**	89.16	89.09	**76.88**	75.40	70.67	87.35	**83.53**	83.47
Consolidation	86.74	85.35	**88.61**	**71.13**	66.52	66.49	**80.83**	72.35	78.98
Edema	**91.57**	90.64	90.60	**73.43**	69.88	68.54	**94.12**	92.65	93.53
Effusion	93.52	**93.73**	91.96	83.15	**83.66**	79.81	**91.40**	91.04	85.94
Pneumonia	80.06	77.73	**83.42**	62.50	59.85	**67.78**	65.20	55.47	**69.91**
Pneumothorax	83.42	81.47	**83.71**	**69.17**	58.69	60.08	68.61	**77.56**	71.76

4. Difference between Fine-tuning performance with encoder freezing and without encoder freezing (ΔFinetune) For this surrogate, we evaluate as a metric the difference between the metrics obtained in surrogates 2 and 3. This new metric represents the loss of fine-tuning performance when we force the pre-trained and final model to use the exact hidden representation. Intuitively, while the surrogate *Finetune* indicates the gained knowledge from tuning the encoder and the decoder, and the surrogate *Freeze* the gained knowledge from tuning the decoder; this last surrogate Δ*Finetune* refers to the gained knowledge from tuning the encoder.

7.3 RQ1: Auxiliary Pathology Learning outperforms single-pathology learning

We compare in Table 7.2 the test (CheXpert → CheXpert) and generalization AUC (CheXpert → NIH and CheXpert → PC) achieved by the best combination of pathologies in our approach (APL), the single pathology case (Single), and the all-pathology case (All).

For the test performance, our approach leads to the best performances for 3/7 pathologies: Atelectasis, Cardiomegaly, and Effusion. Meanwhile, training a single pathology model yields the best test performance only for Effusion, and training on all the pathologies together achieves the best test performances for Consolidation, Pneumonia, and Pneumothorax.

Next, we compare the generalization performance. Our approach achieves the best generalization AUC (CheXpert → NIH) for 4/7 combinations: Cardiomegaly, Consolidation, Edema, and Pneumothorax.

Table 7.3 compares the test performance (CheXpert → CheXpert) and generalization performance (CheXpert → NIH) of different target pathologies when learned in combination with any of the remaining 6. While the standard deviation (Std)

Table 7.3: Statistic of AUC performance computed for different combinations of models trained on the CheXpert dataset and evaluated on CheXpert (left), NIH (middle), and PadChest (right)

	CheXpert → CheXpert				CheXpert → NIH				CheXpert → PadChest			
	Mean	Std	Min	Max	Mean	Std	Min	Max	Mean	Std	Min	Max
Atelectasis	88.83	0.57	87.80	89.59	68.39	0.52	67.73	69.25	68.08	1.29	65.71	69.55
Cardiomegaly	89.21	0.37	88.76	89.71	74.08	1.87	71.90	76.88	85.16	1.67	83.53	87.35
Consolidation	86.12	0.78	84.50	86.74	67.76	2.00	65.04	71.13	75.93	2.83	72.35	80.83
Edema	90.78	0.73	89.43	91.57	64.17	6.06	58.41	73.43	92.75	1.24	91.11	94.12
Effusion	92.93	0.47	92.08	93.52	81.66	1.26	80.15	83.15	89.82	2.17	85.02	91.40
Pneumonia	79.39	0.72	78.10	80.06	58.90	1.93	56.99	62.50	59.84	3.64	55.07	65.20
Pneumothorax	81.85	1.19	80.16	83.42	63.91	3.29	61.21	69.17	65.57	6.07	59.31	77.56

computed on test performance remains low (<1% except for Pneumothorax with 1.19%), the Std jumps up to 6.06 % for Edema. The best performing combination of pathologies with Edema reaches 73.43%, while the worst performing combination with Edema only achieves 58.41%.

Edema and Pneumothorax show the largest variance of performance on target when they are learned with another pathology. Figure 7.1(a,c) show the ROC curves of the source test performance while Figure 7.1(b,d) show the ROC curves of the generalization performance.

Our results show, for example, that the test performance of Edema learned with Consolidation is 1% lower than Edema learned with Effusion. However, the ranking is reversed when considering the generalization performance: Edema with Consolidation gets a 15% higher generalization performance than Edema with Effusion.

7.4 RQ2: Auxiliary Pathology Learning is competitive with SoTA models

Table 7.4 compares our approach's test and generalization performance and ranking and four SoTA CXR classification models.

Across the seven pathologies, our approach ranks second on average for test and generalization performance, behind only the Ensemble Model (EN), which outperforms our technique for all pathologies but one (Cardiomegaly).

Compared to single models, our approach shows higher generalization performance than the DenseNet model (DN) for three pathologies and has similar performance on one pathology (Consolidation, $< 0.1\%$ difference). It outperforms the MixtureNet model (MN) for six of the seven pathologies (all but Cardiomegaly) and outperforms the Fine-tuned model (FT) for all the seven pathologies.

Moving to the approaches' efficiency, we compare in Table 7.5 the required training images for each approach. For each SoTA approach, we provide the

Table 7.4: Comparison of AUC of SoTA models and our approach (APL). The first column denotes the target pathology, and the following columns report Area Under Curve of each model for test performances on Source and Target datasets. *FT* stands for fine-tuning, *DN* for DenseNet, *EN* for EnsembleNet, and *MN* for MixtureNet. In parentheses is the ranking of the approach across the five evaluated approaches.

Main	Test AUC % (Rank)					Generalization AUC % (Rank)				
	DN	EN	MN	FT	APL	DN	EN	MN	FT	APL
(ATE)	90.53 (2)	93.07 (1)	78.34 (5)	86.88 (4)	89.59 (3)	69.95 (2)	71.10 (1)	68.26 (4)	67.10 (5)	69.25 (3)
(CAR)	89.04 (4)	91.00 (2)	92.25 (1)	85.12 (5)	89.71 (3)	73.83 (4)	76.64 (3)	77.47 (1)	67.84 (5)	76.88 (2)
(CON)	87.64 (2)	90.99 (1)	75.77 (5)	84.97 (4)	86.74 (3)	71.20 (2)	74.64 (1)	69.11 (4)	68.56 (5)	71.13 (3)
(EDE)	90.48 (3)	92.62 (1)	78.05 (5)	88.55 (4)	91.57 (2)	71.27 (4)	82.44 (1)	65.82 (5)	72.85 (3)	75.91 (2)
(EFF)	91.22 (3)	95.59 (1)	85.56 (5)	90.84 (4)	93.52 (2)	80.75 (4)	83.71 (1)	80.33 (5)	81.02 (3)	83.15 (2)
(PNE)	84.94 (1)	N/A	65.78 (4)	77.29 (3)	80.06 (2)	63.27 (1)	N/A	62.12 (3)	60.81 (4)	62.50 (2)
(PTX)	82.10 (3)	N/A	82.82 (2)	80.25 (4)	83.42 (1)	73.81 (1)	N/A	59.29 (4)	67.34 (3)	68.17 (2)
Avg Rank	2.8	1.2	4.2	4.2	2.6	3.2	1.4	3.8	4.2	2.4

percentage of images needed by our approach compared to their method. Lower values indicate that our approach is more efficient than SoTA.

Our approach requires between 5.76% and 13.07% of the training set used by MixtureNet. It requires between 24.41% and 55.38% training examples compared to DenseNet and between 27.65% and 62.72% compared to EnsembleNet. We do not compare with Fine-Tuning as this latter requires the same training data as our approach and always has lower performances than ours in Table 7.4.

7.5 RQ3: Guiding the selection of the auxiliary pathology with surrogates

In Table 7.6 we evaluate the generalization performances of single pathology models pre-trained on a source pathology, then fine-tuned on a target pathology. Using Cardiomegaly for pre-training leads to the most significant performance drop for 3/6 pathologies: Atelectasis, Edema, and Effusion. While using Consolidation for pre-training improves generalization performance compared to single pathology training for 3/6 pathologies: Edema, Pneumonia, and Pneumothorax.

In Table 7.7(a), we evaluate the Spearman correlation between the test AUC surrogate metric and our target metric. There are no clear correlations (R=0.06; p-value>5%) between pairs that improve test performance and generalization performance.

We evaluate next the correlation between our target metric and the surrogates obtained by fine-tuning both the encoder and decoder (*Finetune*) or obtained by fine-tuning only the decoder (*Freeze*). We show the results in Table 7.7(b). Fine-tuning a model from the auxiliary pathology to the main pathology has a moderate correlation with the generalization performance of the combination of

Table 7.5: Comparison of Generalization AUC and number of images between SoTA models and our approach (APL). The first column denotes the target pathology, and the following columns report the generalization AUC of each of the models. *DN* for DenseNet, *EN* for EnsembleNet, and *MN* for MixtureNet. The *(% images)* indicates the relative number of images needed by our approach compared to the studied approach. Lower values mean that our approach is more efficient. In bold are the SoTA models that our approach outperforms.

Main Pathology	APL			DN (224k images)		EN (198k images)		MN (950k images)	
	Combination	AUC	# images	AUC	% images	AUC	% images	AUC	% images
Atelectasis (ATE)	ATE + CON	69.25	54.902	69.95	24.48	71.10	27.72	**68.26**	5.77
Cardiomegaly (CAR)	CAR + CON	76.88	54.758	**73.83**	24.41	**76.64**	27.65	77.47	5.76
Consolidation (CON)	CON + EFF	71.13	72.034	71.20	32.11	74.64	36.37	**69.11**	7.58
Edema (EDE)	EDE + CON	75.91	63.438	**71.27**	28.28	82.44	32.03	**65.82**	6.67
Effusion (EFF)	EFF + CON	83.15	72.034	**80.75**	32.11	83.71	36.37	**80.33**	7.58
Pneumonia (PNE)	PNE + CON	62.50	90.009	63.27	40.13	N/A	N/A	**62.12**	9.47
Pneumothorax (PTX)	PTX + PNE	68.17	124.230	73.81	55.38	N/A	N/A	**59.29**	13.07

the main and auxiliary pathologies (R=0.34), while fine-tuning the decoder only shows almost no correlation (R=0.11, p-value>5%).

We evaluate in Table 7.7(c) the correlation between our last surrogate, ΔFinetune, and our target metric. This evaluation yields a high correlation (R=0.47; p-value<5%).

We focus on this last surrogate metric in Figure 7.2. It shows a significant linear regression between both, and we provide its regression function.

7.6 Discussion

Our results demonstrate the key role of pathology interactions when training CXR models on a source population to generalize to a different target population.

7.6.1 Auxiliary Pathology Learning

It is common practice to train CXR classification models with as many pathologies as possible. For example, [CHB+20] trained models on 15 pathologies together, and [IRa19] proposed a benchmark with 11 pathologies. However, our results show that learning all the pathologies is not the best way to achieve robust learning.

For 3/7 pathologies, we can achieve the best test performance by combining all the pathologies; yet, using all the pathologies remains relevant only for 1/7 pathologies when evaluating the generalization performance. Our approach, APL, on the other hand, achieves the best generalization performance for four out of the seven pathologies.

Moreover, not all auxiliary pathologies provide the same improvement in generalization performance. The generalization performance on pathologies like Edema and Pneumothorax is susceptible to the auxiliary pathology we combine during

Table 7.6: (CheXpert → NIH) AUC change with a full model fine-tuning. We train on CheXpert and evaluate on NIH. The model is pre-trained on the auxiliary pathology (row) and then fine-tuned on the main pathology (column). The values are relative changes to the diagonal, where the models are pre-trained and fine-tuned only on the main pathology. In bold is the smallest drop (or highest increase), and in underline, the highest drop.

Main pathology Pre-training pathology	ATE	CAR	CON	EDE	EFF	PNE	PNX
Atelectasis	0.00	-13.77	**5.23**	-0.85	**-2.29**	-1.99	-5.67
Cardiomegaly	-8.85	0.00	3.88	-15.87	-15.27	-0.47	-0.72
Consolidation	-3.86	-9.25	0.00	**9.06**	-4.52	1.04	**7.47**
Edema	-3.46	**-7.44**	2.98	0.00	-12.61	1.79	-2.07
Effusion	**-0.31**	-13.33	4.11	-0.25	0.00	1.40	3.59
Pneumonia	-0.65	-8.05	-2.62	-0.84	-8.79	0.00	-7.50
Pneumothorax	-4.04	-15.42	-7.95	-5.96	-8.18	**5.46**	0.00

	Surrogate Metric	Correlation Coefficient R	p-value
(a)	Test	0.06	0.70
(b)	Finetune	0.34	0.03
	Freeze	0.11	0.45
(c)	ΔFinetune	0.47	1.6e-3

Table 7.7: Spearman correlations between AUC changes of the model and each surrogate metric.

the training. E.g., the Robust AUC of Pneumothorax is as low as 58.41% with Effusion and as high as 73.43% with Consolidation.

Some pathologies, however, always benefit from auxiliary pathology learning. Our evaluation of Pneumothorax combinations exhibits that any auxiliary pathology leads to better performance than the single-pathology model. The worst auxiliary pathology achieves higher generalization performance than the single-pathology learning (61.21% vs. 58.69%).

The first hypothesis behind the difference is the number of positive and negative samples of the auxiliary pathologies and their co-occurrence with the main pathology. However, neither Effusion nor Consolidation has the most positive samples. Similarly, none of them has the most negative samples. Furthermore, they do not have the highest or lowest co-occurrence ratio with Edema. We can draw the same conclusions about all the best and worst combinations of pathologies. (cf Appendix

A. for the detailed ratios).

Next, we hypothesize that the best auxiliary pathologies lead to regularized learning of the neural network. I.e., they share complementary low-level features to the main pathology and hence, improve the robustness of the learning. We analyze the low-level features using the CKA metric. In each of the subfigures in Figure 7.3, we plot the CKA matrix between the 50 layers of each model. Each axis represents the layers of a model. In Figure 7.3(a), we evaluate the CKA similarity from layers of the model trained with Edema and Consolidation (i.e., the model with the highest generalization performance for Edema), and in Figure 7.3(b), the model trained with Edema and Effusion (model with the lowest generalization performance for Edema). Similarly, Figure 7.3(c) and Figure 7.3(d) respectively show the CKA similarity of the model trained with Pneumothorax and Pneumonia (Best generalization performance for Pneumothorax) and Pneumothorax and Edema (Worst generalization performance for Pneumothorax).

In these figures, large block structures (*e.g.*, from layer 22 to 42 in (a) or from layer 0 to 30 in (c)) suggest that these parts of the models rely on similar representations. On the contrary, smaller blocks (*e.g.*, centered around 10, 25, or 40 in (b) or centered around layer 8, layer 17, or layer 32 in (d)) suggest that smaller portions preserve stable activations through the network. These patterns are also present in other combinations of pathologies (see Appendix D). Overall, larger blocks appear in highly generalizable models, while strides appear in less generalizable ones. In addition, previous research suggested that large block structures primarily appear in over-parameterized models [KNL+19] which may explain why these combinations of pathologies lead to models with higher generalization performance.

This qualitative study confirms that there are patterns in the low-level features that contribute most to the generalization performance of pairs of pathologies

7.6.2 Auxiliary Pathology Learning is competitive with SoTA models

Our approach outperforms all SoTA single models for CXR and remains competitive against ensemble models for Cardiomegaly and Effusion while requiring only one-third of the data. In general, our results demonstrate that mixing many pathologies (DenseNet and EnsembleNet) or multiple datasets (model MixtureNet) does not improve the generalization performance and only leads to expensive training. The MN model uses up 17 times more data than our approach and leads to less generalizable models for six of the seven pathologies we evaluate.

These results reinforce the need for cheap techniques to identify the best auxiliary pathologies, as adding more data with more datasets remains less efficient than selecting the right auxiliary pathology.

While our approach is designed to minimize the training data collection process and cost, it can be used in combination with the strategies proposed in DenseNet, EnsembleNet, or MixtureNet when more data are available. Designing optimizations over the datasets or the pathologies to include and the ensemble to build is a natural follow-up of this work.

7.6.3 Guiding the selection of the auxiliary pathology with surrogates

Our results confirm the insights from the previous sections: Firstly, test performance is a misleading metric for selecting the best combination of pathologies. There is no concrete evidence that the combinations of pathologies that perform the best on one's test dataset also perform the best on an external dataset.

Next, the interactions between the pathologies happen most probably at the low-level features, and combinations that lead to the most significant change in test performance between fine-tuning with encoder-freezing and without encoder-freezing also lead to the most significant change in generalization performance.

Finally, when we rely on pre-training and fine-tuning, selecting the right starting pathology in the pre-training can lead to improved generalization performance. While no auxiliary pathology works for all, Cardiomegaly seems to perform the worst on average, and Consolidation the best on average.

7.6.4 Other factors that affect generalization

While our research investigates the impact of pathologies on generalization, we concede that our findings can be affected by other confounding factors (*e.g.*, ethnicity or age). However, compared to single pathology learning, our approach mitigates the drop in performance when we deploy the models on a dataset with different age and sex distributions. For example, for Pneumothorax, the AUC drops from 83.42% to 69.17% with our approach and to 58% with single pathology learning. Follow-up work could evaluate the links between age, gender, capture process, and pathology interactions but is out of the scope of our study that focuses on training generalizable models with limited data.

Building safe models for hospital settings entails several maintenance considerations. Such as monitoring the deployed models for potential risk-sensitive events, *e.g.*, data drift derived from population changes, different distribution of features, new commodities, and other public health events. Such changes may require re-calibration or re-training. However, knowing the source and target distributions requires not only knowledge about the features but also the metadata (e.g., which devices were employed for data acquisition). Therefore, the life-cycle of a deployed model is not limited to sustained model work but also entails constant data work. Data collection, especially in medical settings, entails heterogeneous data sources

that may hamper the effective collection of representative datasets and increase the overall operational expenditure. Such associated high costs become hard to overcome in developing countries [SK21]. For these reasons, it is crucial to develop tools for effective evaluation and guidance of such processes.

Our focus on CXR is motivated by the fact that respiratory infections are the leading cause of death in developing countries [FS14]. It is particularly hazardous for children, the geriatric population, and immunocompromised patients, causing over 15% of deaths in children under five years old worldwide [RKA17].

We believe that using ML systems for CXR diagnosis has the most impact on under-serviced populations that have limited access to specialists. They are, however, also the most under-represented in the datasets used to build and benchmark these ML systems. Some of the most extensive datasets are from NIH (NIH-14, USA), Stanford Hospital (CheXpert, USA), and San Juan Hospital (PadChest, Spain), which cover similar Caucasian populations and pathology distributions. Even if these datasets have close origins, our results show a drop of about 20% AUC moving from one dataset to another.

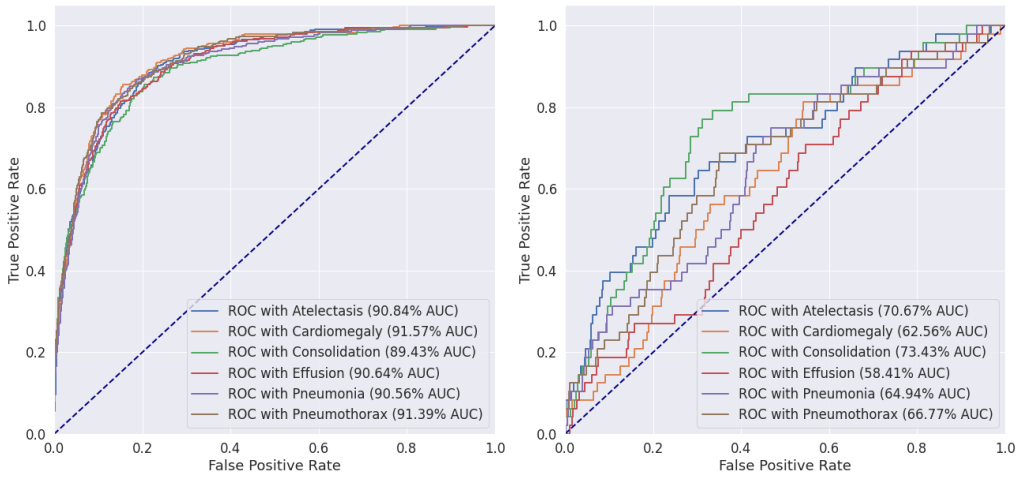
Our work also shows that data collection and annotation can be optimized by focusing on a subset of pathologies to improve generalization. The choice of pathologies can be driven by empirical evaluations (as proposed in our approach) supported by domain knowledge provided by practitioners.

7.7 Conclusion

This work focuses on the critical topic of generalizable ML systems for medical diagnosis. Ensuring that the systems that have been designed and tested by the research community are effective on very different populations is of utmost importance before we consider deploying them in practice.

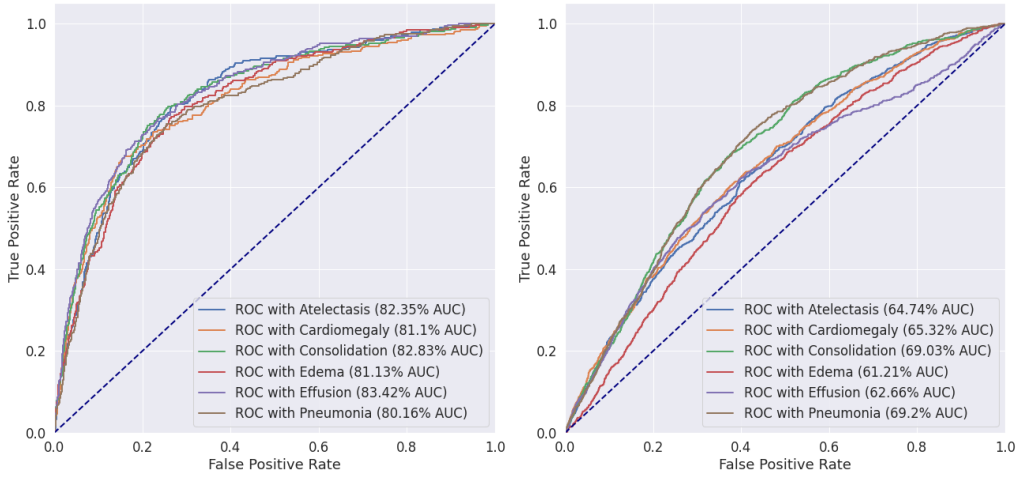
One common practice to improve generalization performance is to launch large data collection campaigns to build datasets for training and fine-tuning across many pathologies. However, our study demonstrates that a better understanding of the pathologies and combination of only **two** pathologies can lead to models that require a fraction of the data and still manage to outperform models trained with large datasets, ensembling, or fine-tuning. Our approach, combined with our surrogate metrics for pathologies selection, is among the most effective and efficient ways to provide reliable and cost-effective ML systems for medical diagnosis, especially for populations with limited medical facilities and resources.

Following our research, we advise ml practitioners to dedicate time and resources to understanding the interactions between pathologies in target populations instead of building larger, complex, and data-hungry models to tackle ML-based medical diagnosis. Our research, in a nutshell, champions the saying: *More data is good, smart data is better.*



(a) Test - Edema

(b) Generalization - Edema



(c) Test - Pneumothorax

(d) Generalization - Pneumothorax

Figure 7.1: ROC curves of source performance (CheXpert→CheXpert) and target performance (CheXpert→NIH) for edema (top) and pneumothorax (bottom) when learned with the 6 other pathologies.

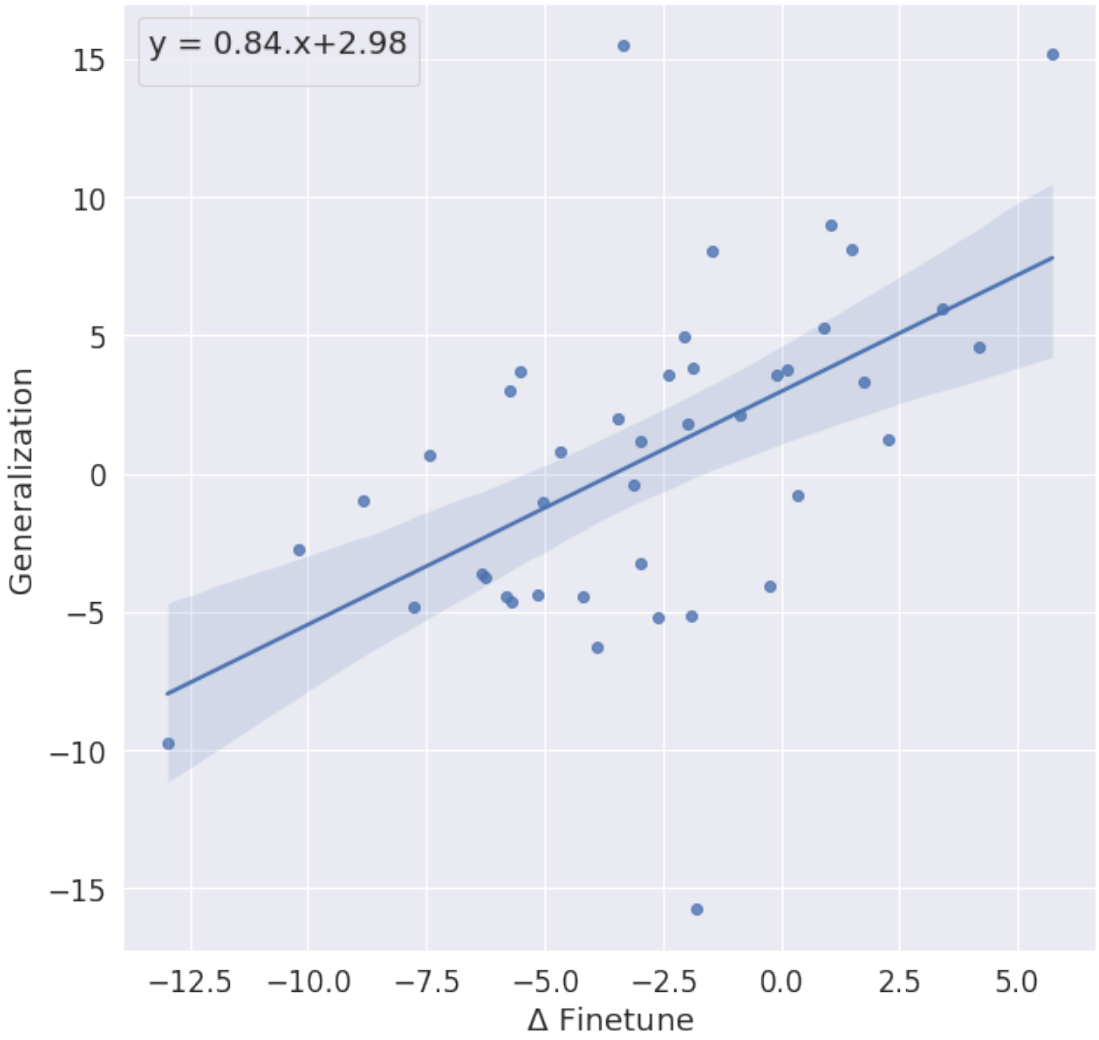
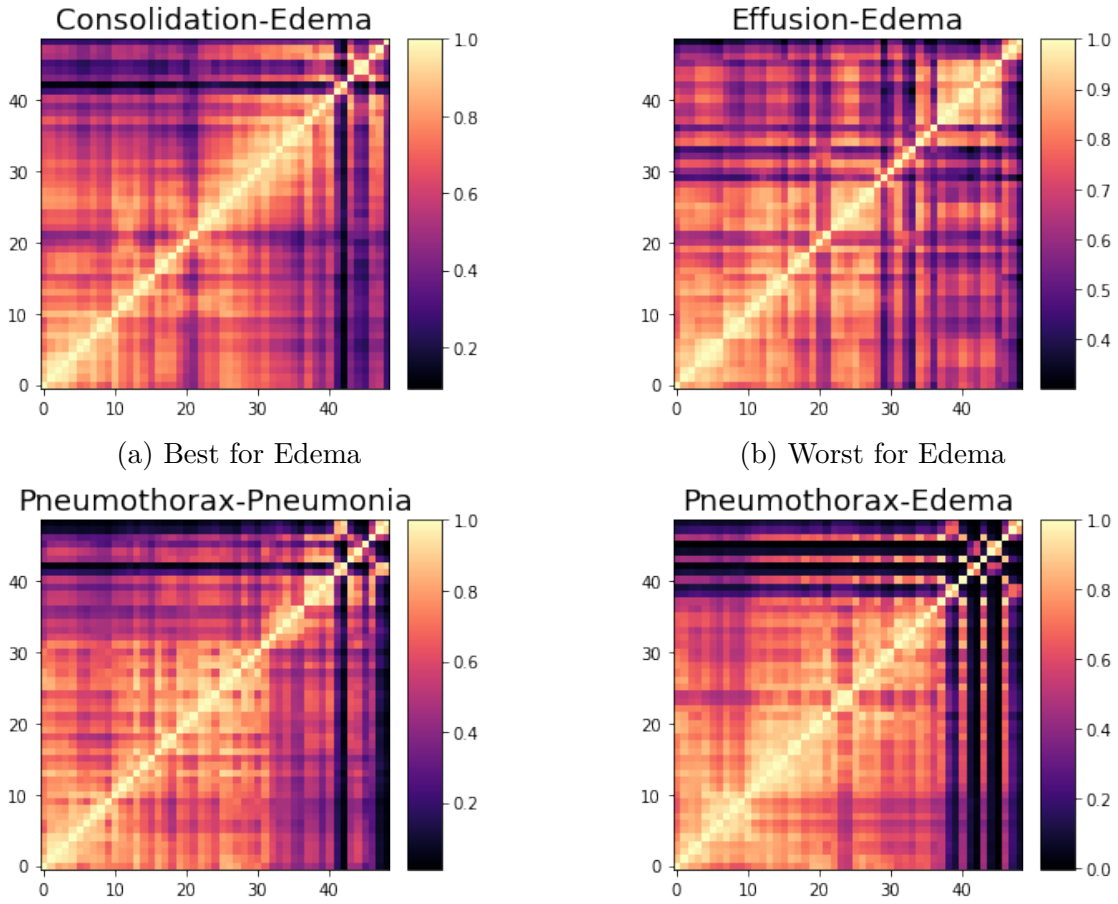


Figure 7.2: Linear regression between Chex→NIH Generalization AUC and the change of performance between fine-tuning and encoder-freezing across all pairs of pathologies.



(a) Best for Edema

(b) Worst for Edema

(c) Best for Pneumothorax

(d) Worst for Pneumothorax

Figure 7.3: Layer Similarity for best (left) and worst (right) generalization performances (Chex→NIH). The main pathology of the top row is edema and the main one for the second row is Pneumothorax.

8

Data-driven Simulation and Optimization for Covid-19 Exit Strategies.

The rapid spread of the Coronavirus SARS-2 is a major challenge that led almost all governments worldwide to take drastic measures to respond to the tragedy. Chief among those measures is the massive lockdown of entire countries and cities, which beyond its global economic impact has created some deep social and psychological tensions within populations. While the adopted mitigation measures (including the lockdown) have generally proven useful, policymakers are now facing a critical question: how and when to lift the mitigation measures? *A carefully-planned exit strategy is indeed necessary to recover from the pandemic without risking a new outbreak. Classically, exit strategies rely on mathematical modeling to predict the effect of public health interventions. Such models are unfortunately known to be sensitive to some key parameters, which are usually set based on rules-of-thumb. In this chapter, we propose to augment epidemiological forecasting with actual data-driven models that will learn to fine-tune predictions for different contexts (e.g., per country). We have therefore built a pandemic simulation and forecasting toolkit that combines a deep learning estimation of the epidemiological parameters of the disease in order to predict the cases and deaths, and a genetic algorithm component searching for optimal trade-offs/policies between constraints and objectives set by decision-makers. Replaying pandemic evolution in various countries, we experimentally show that our approach yields predictions with much lower error rates than pure epidemiological models in 75% of the cases and achieves a 95% R^2 score when the learning is transferred and tested on unseen countries. When used for forecasting, this approach provides actionable insights into the impact of individual measures and strategies.*

Contents

8.1 Introduction

Since the outbreak of the COVID-19 pandemic, the world has been facing a human tragedy with overwhelmed healthcare systems and fears of economic collapses. In the absence of vaccines to immunize the population rapidly at scale, governments have implemented various non-pharmaceutical public health interventions such as social distancing and lockdowns. Considering that the World Health Organisation (WHO) is foreseeing first clinical trials of vaccine for the end of the year 2020 [Wor20], decision-makers must carefully plan their exit strategies: measures that were put in place to contain the coronavirus spread must be methodically lifted to avoid the risk of precipitating new outbreaks.

In this context, mathematical modelling offers public health planners with frameworks to make predictions about the spread of emerging diseases and assess the impact of possible mitigation strategies. This is particularly important when dealing with infectious diseases, such as COVID-19, where mass interventions (*e.g.,* screening, social distancing, and vaccination) can lead to effects at a population level, including herd immunity, changes in the infection rate or even changes in the pathogen ecology as a consequence of selective pressure.

There are two main types of models: static cohort models and transmission dynamic models [JB11]. Static models, typically relying on decision trees and Markov processes, assume a force of infection that is independent of the proportion of the population that is infected and therefore is of little use in response to highly infectious diseases like COVID-19. Transmission dynamic models, on the other hand, see a force of infection varying depending on the proportion of the population which is infected. Compared with static cohort models, transmission dynamic models are usually more complex to parameterize requiring epidemiological information on the infectious disease and demographic and economic information about the affected population.

Different techniques exist to implement dynamic approaches. Agent-Based Models (ABM) are simulations composed of agents that interact with each other and their environment. Because each agent can make its own rules, this type of approach can capture aggregate phenomena derived from the behavior of single agents. These models offer a great explainability of the root causes leading to the propagation of a disease but are computationally intensive to run and thus, hardly applicable to large populations. Indeed, the behaviour and the interaction of each type of agent needs to be fully defined in order for the model to be useful. These rules are case-specific and are not transferable from one population to another.

The most common approach to model the spread of infectious disease is the Susceptible-Infected-Removed (SIR) model and its extension *i.e* SEIR (Susceptible, Exposed, Infectious, and Recovered). This is a state-based model, every state expresses the degree of exposure of a population to the disease. It is also equation-

based where each equation defines the rate to go from one state to the other. The SEIR model thus separates the population into four groups and simulates the evolution over time of each one of the subpopulations. The transition rates are defined by the time scale to which an individual can transmit the disease, the time to recovery, and the number of newly infected people due to an infected individual. The most varying parameter is effective reproduction number (R_t), and expresses the number of people that can be contaminated by an infectious individual over a period of time.

These methods are dependent on the validity of the input parameters like transition rates. While SEIR is a very powerful model, it presents a major limitation, it requires hyper-parameters that are hard to observe such as the infection rate of an individual. In practice, SEIR parameters are manually set to fit with the local observations to the considered population (e.g. country), and are not learnt from larger-scale observations. To circumvent the limitation of such epidemiological models, researchers started to take advantage of the advances made in Machine Learning (ML) in order to create models based on available large datasets [VMU+20b; SCC+20a]. We name this family of approaches "ML-based". and our own work falls in it.

Our first contribution is to devise a novel approach, *DN-SEIR*, that alleviates manual tuning of the SEIR model, by relying on large and trustable public datasets (large scale observations) and machine learning (to learn the parameters' values for a given population). Our approach combines SEIR with a machine learning predictor, based on a deep learning model, to estimate the effective reproduction number (R_t), over time. The machine learning predictor relies on demography and mobility features to predict an effective reproduction number. For each time increment, R_t is updated and used for the next day computation. We evaluated our approach on twelve countries from all continents and showed that our approach that mixes demographics, mobility, and epidemiological data provides better forecasts for 9 out of 12 of the studied countries than a purely epidemiological modelling.

Our second contribution is to exploit this online prediction of effective reproduction number in a simulation tool[1] for policymakers which was recently advertised to the public [2]. Policymakers have to decide when to relax certain parts of society (workplaces, travels, schools...) and to what extent it may create a new epidemic wave that would flood the hospitals with critical cases. The simulator enables one to make such a strategic exit plan for a certain country and predict its impact in terms of hospitalization, infected people and deaths. It is also designed to explore and optimize various exit strategies and constraints. We evaluate 3 common handcrafted exit strategies and show that multi-objective genetic algorithms can find

[1]Open sourced and available on https://github.com/yamizi/Covid19

[2]https://t.co/FN5pn1dMOR

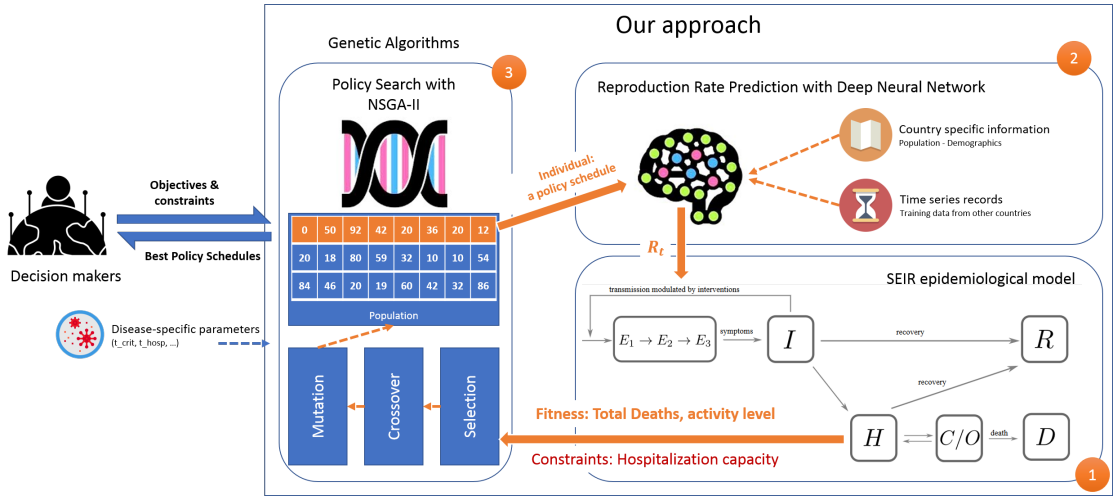


Figure 8.1: Our approach relies on a feedback loop where the Genetic Algorithm searches for optimal exit strategies using a fitness function computed from the epidemiological model outputs. The epidemiological model's parameters are learned with a Machine Learning algorithm that uses population mobility behaviours and demographics as input features.

atypical strategies on the pareto-front that minimize both the death numbers and the economic impact.

8.2 Approach

Our end goal is to provide policymakers with a tool to easily generate exit strategies and evaluate their impact. In particular, an exit strategy can be modeled like as a schedule of measures (*policy schedule*) that will impact the way the disease will spread. We restrict the policy schedule to mobility levels.

As illustrated in Figure 8.1, we propose to combine a genetic algorithm (to search for policy schedules), a deep learning model (to predict the evolution of the effective reproduction number induced by a given policy schedule) and an epidemiological model (to forecast, based on the computed effective reproduction numbers, the effect of the scheduled policies on public health over time, e.g. deaths and hospitalization occupancy). Our three components work within a feedback loop. At each iteration, the genetic algorithm builds a population of policy schedules, which the deep learning and epidemiological models allow to evaluate. In turn, this feedback is used to generate better schedules, optimizing health-related objectives (e.g. minimize total deaths) while satisfying hard constraints (e.g. never exceed hospitalization capacity).

8.2.1 Estimating Impacts on Public Health with Epidemiological Models

Epidemiological models predict the state of a population struck by a pandemic over time, based on state transition parameters and the evolution of the effective reproductive number, R_t, of the disease.

We use an extension of SEIR model, i.e. the SEI-HCRD compartmental model – Susceptible (S) → Exposed (E) → Infectious (I) → Removed (Hospitalized (H), Critical (C), Recovered (Rec), Dead (D)). Such model can be defined by the following system of ordinary differential equations:

$$\frac{dS}{dt} = -\frac{R_t}{t_{inf}} \cdot I \cdot S \tag{8.1}$$

$$\frac{dE}{dt} = \frac{R_t}{t_{inf}} \cdot I \cdot S - \frac{1}{t_{inc}} \cdot E \tag{8.2}$$

$$\frac{dI}{dt} = \frac{1}{t_{inc}} \cdot E - \frac{1}{t_{inf}} \cdot I \tag{8.3}$$

$$\frac{dH}{dt} = \frac{1-m}{t_{inf}} \cdot I + \frac{1-f}{t_{crit}} \cdot C - \frac{1}{t_{hosp}} \cdot H \tag{8.4}$$

$$\tag{8.5}$$

$$\frac{dC}{dt} = \frac{c}{t_{hosp}} \cdot H - \frac{1}{t_{crit}} \cdot C \tag{8.6}$$

$$\frac{dRec}{dt} = \frac{m}{t_{inf}} \cdot I + \frac{1-c}{t_{hosp}} \cdot H \tag{8.7}$$

$$\frac{dD}{dt} = \frac{f}{t_{crit}} \cdot C \tag{8.8}$$

such that $S + E + I + H + C + Rec + D$ is equal to the total population and R_t denotes the effective reproduction number over time.

The SEI-HCRD involves several parameters. t_{suffix} is the transition time estimated to transit from one population to the other. t_{inc} is the average incubation period, t_{inf} is the average infectious period, t_{hosp} is the average hospitalization time in normal state (i.e. until any patient recovers or enters a critical state). t_{crit} is the average hospitalization time in a critical state (i.e. until death or recovery). The parameters m, c, f determine the severity of the infection: m is the percentage of infected individuals with non-severe symptoms (i.e. they are asymptomatic or have mild symptoms) and which, therefore, are not hospitalized. c is the percentage of hospitalized persons who will eventually enter a critical state. Finally, f denotes the percentage of persons in the critical state who will pass away.

Transition time	Value	Transition ratio	Value
t_{inc}	5.6 days	m	80%
t_{inf}	2.9 days	c	10%
t_{hosp}	4 days	f	30%
t_{crit}	14 days		

Table 8.1: Parameter values used in the SEI-HCRD model.

To set these parameters, we lean on Liu et al.'s study [LGW+20] and assign them with the constant values reported in Table 8.1. Then, given the time series of effective reproduction numbers over time,$\{R_t\}$, the SEI-HCRD model computes the resulting impacts on the population, including the number of deaths. Instead of manually assigning fitted values to R_t, we propose to predict them from scheduled exit strategies using deep learning models.

8.2.2 Predicting the Effective Reproduction Number over time with Deep Learning

Feature engineering

We start from Google's Mobility Reports[3], which track the mobility trends over time, for different categories of places in 97 different countries. Each feature corresponds to a category of places and its value captures the daily traffic of such places. More precisely, the value of the feature is the difference between the daily traffic and the traffic baseline (i.e. the median traffic for the same weekday during the 5-weeks ranging from January 3 to February 6, 2020). The reports include 6 categories: Grocery & pharmacy, park, transit stations, retail & recreation, residential and workplace.

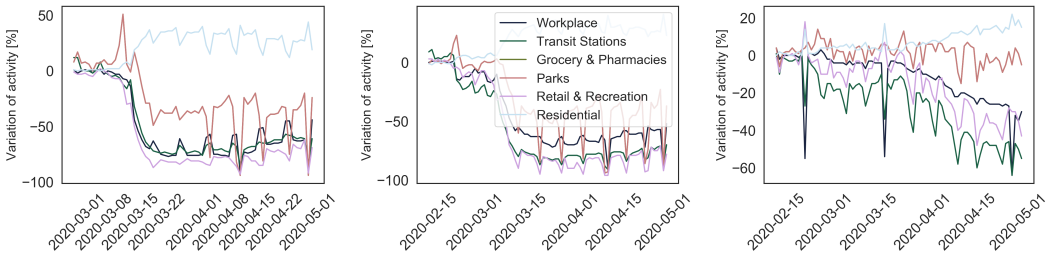
Of course, the values of these features are country-specific and are largely impacted by the mitigation strategy of each country. For example, Figure 8.2 shows the evolution of all features for Luxembourg, Italy and Japan. We observe that Italy and Luxembourg have drastically reduced their activities whereas Japan does not exhibit a significant reduction, for the case of schools and international travels.

Next, we clean the collected data. When some values (for a given feature) are missing, we fill the gap by interpolating between the closest days with available information. For each category of places, we smooth the corresponding feature over the 5, 10, 15, and 30 past days, resulting in four new features.

We complete our dataset with demographic features and with the corresponding day of the week to take into account the weekly fluctuations of the data.

Overall, our feature engineering process yields 4,625 inputs of 32 features each,

[3]https://www.google.com/covid19/mobility/



(a) Luxembourg (b) Italy (c) Japan

Figure 8.2: Evolution of the mobility indicators for Luxembourg, Italy, and Japan. A value of 0 means that the activity is at the same level as before the confinement, a value of -100% is a total stop of the activity and a positive values shows an increase of the activity compared to the reference value.

which are recapitulated in Table 8.2.

Training & validation

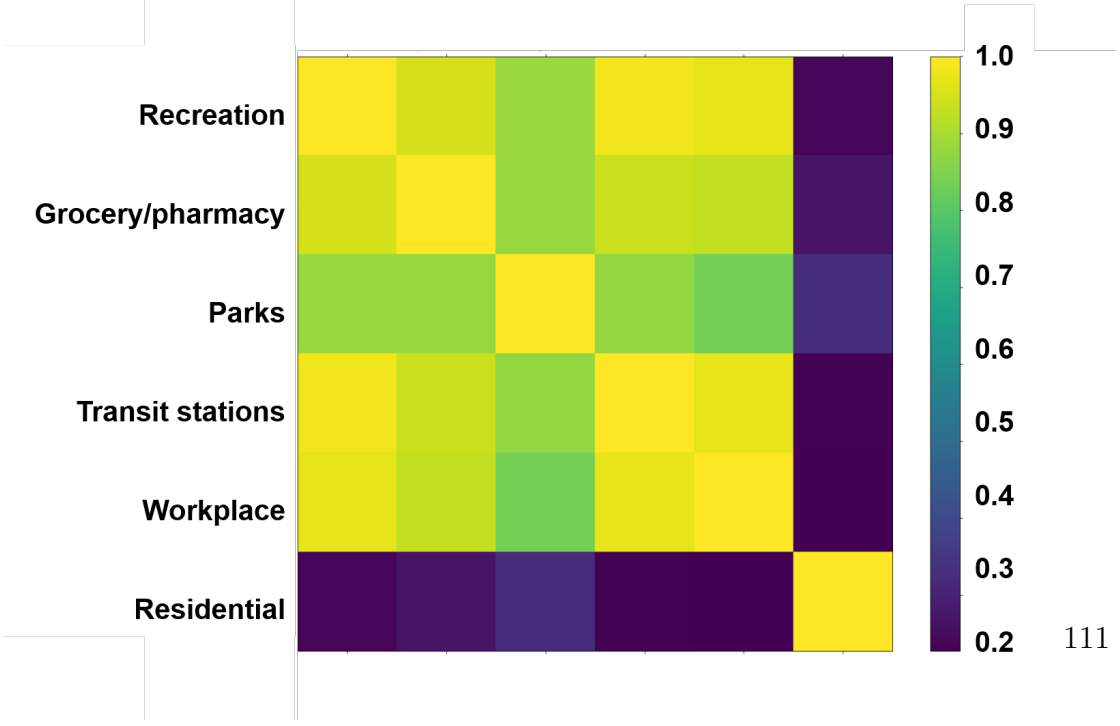
The model can be seen as a supervised predictor, taking as input the mobility and demographic features to predict an effective reproduction number, R_t, for each time index t. Thus, in order to train it, we need to label the dataset with R_t value for each day of the training period.

Since the real effective reproduction numbers are not known, we estimate them by fitting the SEI-HCRD model to the real-world number of cases and deaths. Since the R_t are time-dependent, we represent them as a decaying function and seek the best parameters' value for this function (which yield the fittest SEI-HCRD model). We opted for the Hill decay function because it showed good results in the literature [SMR+08]. Thus, R_t is given by $\frac{1}{(1+(\frac{t}{L})^k)} \cdot R_0$ where R_0, k and L are parameters. We use the L-BFGS optimization method to find the values of these parameters that minimize the Mean Square Error of the number of cases and deaths predicted by the SEI-HCRD model. Once these parameter values are found, they can be injected back in the Hill decay function to generate a time series for the past values of R_t.

The analysis of feature correlations (Figure 8.3a) shows that when working on a country-by-country basis, all the mobility features are highly correlated (over 0.75) for some countries as all the activity reduction and closure were enacted in most sectors at the same time. Co-linear features offering little to no information gain to the learner, we train the model on all countries to reduce any correlation between the different features. Thus, the model is trained using all the countries at once and the train/test split is done randomly.

Once a training set with expected values (R_t values computed from the estimated

(a) Luxembourg



(b) All Countries

Figure 8.3: Mobility feature correlations.

Feature category	Feature	Values
Mobility	Grocery & Pharmacy	[-100,50]
	Parks	[-100,50]
	Residential	[-100,50]
	Retail & Recreation	[-100,50]
	Transit stations	[-100,50]
	Workplace	[-100,50]
Demographics	GDP	$\mathbb{N}$
	Population	$\mathbb{N}$
	Density	$\mathbb{R}$
	Area	$\mathbb{N}$
	Proportion of population under 15yrs	[0,1]
	Proportion of population over 64yrs	[0,1]
	Day of week	[0-6]
	Region (Continent)	[0-10]

Table 8.2: Features of the Feed Forward Neural Network. Mobility features are augmented by smoothing over 5, 10, 15 and 30 days, hence each mobility feature corresponds to 4 inputs.

Hill decay function) is established, a supervised model can be trained to predict the future values of R_t using the mobility and demographic data as features.

To do so, we rely on a Feed-Forward Neural Network (FFNN). The architecture of the FFNN and its hyper-parameters are optimized using a grid search to minimize the mean square error with cross-validation. The search leads to an architecture with 2 fully-connected hidden layers with, respectively, 1000 and 50 neurons.

In addition to the FFNN, we also evaluate two other estimators. Since the problem takes the form of a time series, we investigate the performances of a Long Short-Term Memory (LSTM) network which allows sequence to sequence transformation, hence, learning from the features of the past days to predict the next days. We use a 15 days window to predict the values for the next 7 days. Note that in this case, at each step, we use the computed values of R_t from the past iterations as input for the model in addition to the rest of the features. Finally, we investigate one last approach, Gradient Boosting using 500 estimators. For each of the approaches, we evaluate the performances using two classical metric, *i.e* the coefficient of determination, (R^2 score) and the Root Mean Square Error (RMSE)[4]. The train:test splits were performed with (1) random split between train

[4]The closer R^2 score to 1 the better and the closer RMSE to 0 the better

and test data (2) split based on the region the country is located in (i.e testing on unseen countries) (3) split based on the time, all values before a certain date are considered for the training set and the ones after are used to build the test set. In all instances, we keep a ratio train:test around 80:20.

The FFNN provided a R^2 score of 0.95 with a random split and 0.97 with a region split, while the gradient boosting could only achieve a R^2 score of 0.83. The LSTM offered slightly better performances with a R^2 score of 0.95 when splitting on a region base to over 0.99 when splitting on a time base. We use the FFNN model in our following experiments. We refer to the combination of FFNN and SEI-HCRD as **DN-SEIR**.

Interpretability

Although Machine Learning algorithms provide increased accuracy in a wide variety of domains, their black-box nature makes them inherently non-interpretable. Indeed, in our case, a multi-layer neural network solely contains information in the form of numerical weights and connections. The model reasoning from input to output remains opaque. Nevertheless, interpretability can be reached as a post-hoc analysis through an independent interpretability framework. We choose Shapley Additive exPlanations, *SHAP* [Mol19] as it provides intuitive visualization-based explanations which can be incorporated in the simulator. The *SHAP* framework is based on game theory and Shapley value. In game theory, Shapley values indicate how to fairly distribute a 'payout' among players. A model can be thought of as a game where each feature value for each instance is a player, while a prediction or model output is a payout. In practice, the Shapley value of a feature value is the contribution of this value for this particular exemplar compared to the average prediction for the specific dataset. *SHAP* provides several advantages. First, this framework has different types of explainers to optimally provide explanations to different models, whether tree-based or kernel-based. Moreover, *SHAP* exhibits an Efficiency property through the Shapley values. Indeed this component guarantees that the difference between prediction and average prediction is fairly distributed among the values of the features of this particular prediction.

8.2.3 Optimization of Policy Schedules with Genetic Algorithms

Our search method uses NSGA-II [DPA+02], an established Genetic Algorithm (GA) for multi-objective optimization that uses non dominated sorting to find pareto-optimal solutions.

Solution space: Any solution generated by NSGA-II is a policy schedule. A schedule consists of a list of vectors, each of which is associated with a mobility feature and encodes the value of the feature for each time index t. A value ranges

from 0 (no restriction) to 100 (full lockdown). The indices t go from April 30 to September 30, with steps of 2 weeks.

Objectives: We use 2 fitness functions which represent the compromising health and societal impacts of policy scheduled. We quality these impacts with the total number of total deaths between April 30 and December 30 and the mean of the mobility feature values over the same period. The first objective must be minimized while the second is maximized.

Constraints: For a policy schedule to be an acceptable solution, we require that the number of critical cases never exceeds the hospitalization capacity (ICU) of the country of interest. This is an important requirement for policymakers, as critical cases which may not be correctly hospitalized likely result in additional deaths.

Selector: Current Pareto-front solutions are selected in priority. If there are more than the population size, they are filtered based on a crowding distance (here, we use Manhattan distance in the objective space). Otherwise, we fill the population with non-optimal solutions, selected using a binary tournament selection: Pareto-dominant solutions are retained in priority; in case of non-dominance, crowding distance to the Pareto front is used for tie-breaking.

We rely on Pymoo [5] to implement our NSGA-II search and use the library's default values for the remaining parameters like mutation rate or crossover.

8.3 Research questions

Our end goal is to provide decision-makers with a tool allowing to easily generate exit strategies in order to evaluate their impact. To achieve this, we use a deep neural network as a proxy to evaluate the hyper-parameters of a SEI-HCRD model based on mobility and demographic data. However, to be useful, the neural network needs to be able to capture enough information in the mobility and demographic data alone to make accurate predictions. Hence, we formulate the first research question as follow:

> **RQ1:** Can we predict the effective reproduction number based on mobility and demographic data?

Ultimately the proposed approach is intended to allow policymakers to evaluate and select exit strategies by analysing their impact on multiple aspects such as the number of death, possible overflow of healthcare capacities or the perturbation of economic activities. To evaluate the capacity of our approach to model such strategies, we evaluate it by predicting the impact of various popular scenarios. Thus, we ask the following question:

[5] www.pymoo.org, the most starred python GA library on Github

RQ2: How does our approach react under different exit strategies?

We conclude our investigation by a comparison of the impact of the popular scenarios with the impact of the one proposed by the search algorithm. The algorithm minimizes the number of deaths and the socio-economical impacts generated by a diminution of activities (mobility) while avoiding the over-saturation of healthcare capacities. To evaluate the results, we compared them to the "naive" scenarios formulated in the previous research question and thus ask:

RQ3: How do the exit strategies proposed by the search algorithm perform against popular ones?

8.4 Results

8.4.1 Predicting the Effective Reproduction Number

Comparison of a fitted SEIR model and our DN-SEIR model We estimate the confidence interval by evaluating the mean and standard deviation of a Bayesian Ridge Regressor on each element of the test set (using the same training set as the FFNN). We use a grid search over 3 values for each of its 2 hyper-parameters α_{init} (0.1, 1, 1.9) and λ_{init} (1, 0.1, 0.01).

We compare in Table 8.3 the predicted cases of the time-dependant SEIR (Fitted on past cases/deaths) and the DN-SEIR approach. The DN-SEIR approach has a lower error to the ground truth values of cases for 9 over 12 countries in comparison with a time regression approach. Besides, the ground truth falls within the confidence interval of the DN-SEIR model for 10 of the 12 countries evaluated. It is worth noting that while 5 of the 12 countries lie under 5% error, 9 over 12 countries do no exceed 15% error.

In Figure 8.4, we evaluate the RMSE between the predicted cases of the SEIR model and the DN-SEIR approach (with its optimistic and pessimistic boundaries) across all the countries of the dataset. The simulation spans on 7 days (instead of 12 days for the results presented in table 8.3) and we compare the number of cases on April 29th, the last day of available mobility data for all countries. We use the Wilcoxon signed-rank test to compare if two distributions are equals. The results show that for all three prediction DN-SEIR, DN-SEIR max and DN-SEIR min we can reject the hypothesis ($p\text{-}values \ll 0.05$) that they are equal to the SEIR prediction. We then perform a Vargha and Delaney's A_{12} test to analyze the effect size. We see that our approach generates a lower RMSE with a small effect size. These results indicate that even in very short term (7 days), relying only on the mobility data yields better results than applying a regression over the past values.

Model interpretation at the global level We fit *SHAP* on the full dataset and show in Figure 8.5 the summary of its impact analysis where the features are

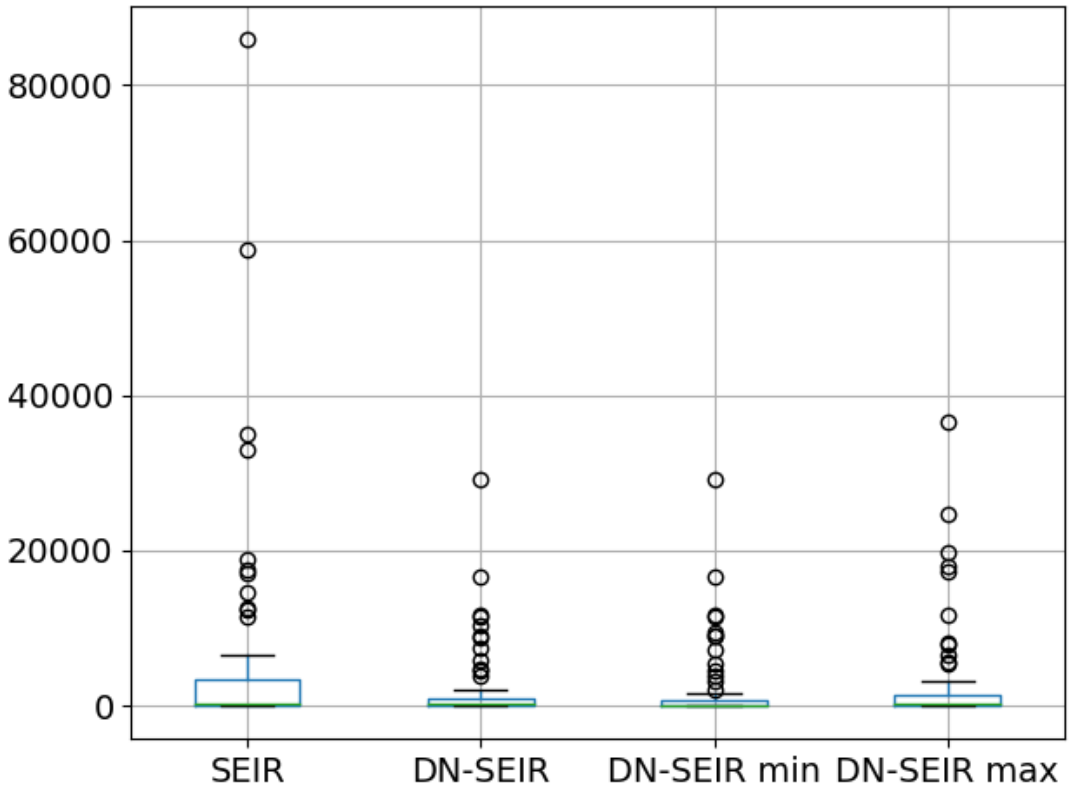


Figure 8.4: RMSE between true cases and predicted cases by our DN-SEIR model and a fitted SEIR model.

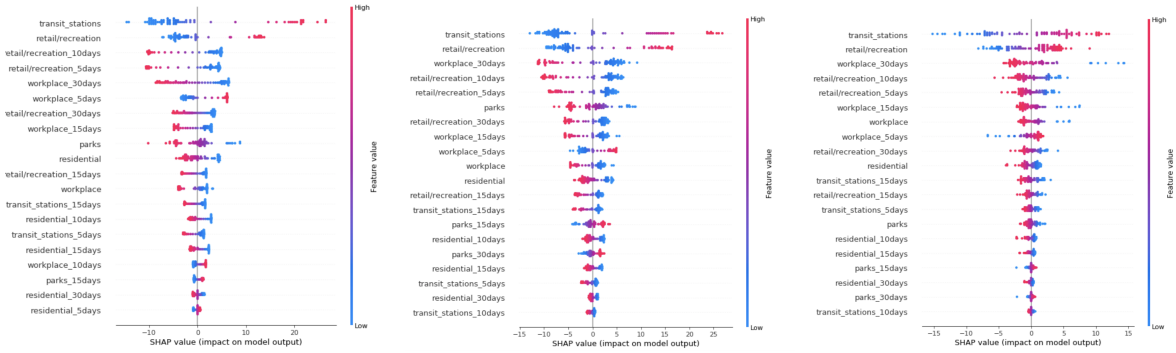
Country	SEIR cases	DN-SEIR cases	True cases	ϵ_r	ϵ_{rSEIR}
Belgium	47,219	51,532 [45,130-60,464]	47,859	0.07	0.01
France	213,103	191,188 [166,043-224,554]	128,442	0.33*	0.66
Germany	161,383	181,438 [163,333-207,151]	157,641	0.13	0.02
Greece	2,429	2,478 [2,343-2,666]	2,576	0.03*	0.06
Italy	6 201,802	202,369 [187,205-223,194]	203,591	<0.01*	<0.01
Luxembourg	3,668	3,724 [3,578-3,936]	3,769	0.01*	0.03
Spain	209,646	230,794 [211,299-257,782]	240,743	0.04*	0.13
Brazil	60,714	68,271 [55,268-86,164]	78,162	0.14*	0.22
Cameroon	1,432	1,645 [1,375-2,003]	1,832	0.11*	0.22
Canada	77,614	63,520 [51,977-78,913]	51,597	0.19*	0.5
Japan	12,250	11,353 [10,010-13,288]	14,088	0.24	0.13
United Kingdom	201,701	160,442 [134,251-188,088]	165,221	0.03*	0.22

Table 8.3: Total cases as of 29/04 as predicted by a time-regression SEIR, and by DN-SEIR model. Both models trained until 11/04. ϵ_r and ϵ_{rSEIR} are the absolute relative error of the DN-SEIR model and time-regression SEIR respectively. (*) indicates that the DN-SEIR yields less error than the time-regression SEIR.

ordered in decreasing order of influence. The distribution of each feature spans horizontally to inform on the impact of the feature on the final decision (positive on the right side), while the colour of each point of the distribution provides information on the range of values of the feature that have an impact (redder color codes for higher values of the feature). We can see for instance for the feature **transit station** that higher values have a positive impact on prediction, i.e higher values of **transit station** translate into higher values of R_t. This insight is common across the three countries. The same goes for the features related to **retail and recreation** activities.

SHAP shows how the trends in the feature, modelled in our approach using smoothed features, play a significant role in the final prediction, and shows an opposing contribution to its associated daily feature. For transit, retail, and park features, the trends values over 5, 10 and 15 days counter the impact of their respective daily values, yet on a lower scale. This indicates an inertia phenomenon that can be explained by the actual delay between the actual numbers (R_t, cases, deaths) and the reported one and also the delay inherent to the epidemiological model.

The comparison of the three countries also shows that mobility features have a much higher impact on the prediction in Italy and Luxembourg than Japan as their SHAP impact is much wider. This hints that other social distancing features in Japan could reduce the impact of mobility (masks for instance).



(a) Luxembourg (b) Italy (c) Japan

Figure 8.5: Global interpretation of the model for Luxembourg, Italy and Japan.

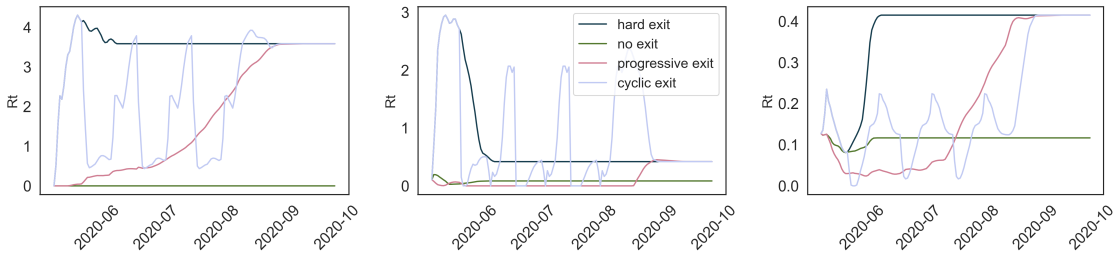
RQ1 Answer: Our approach yields predictions with much lower errors than pure epidemiological models in 75% of the cases and achieves a 95% R^2 score when the learning is transferred and tested on unseen countries.

8.4.2 Mid-term predictions with exit strategies

In this section, we investigate four prediction strategies for an exit from the lockdowns that were taking place all over the world. The goal of the exit strategies is to allow a return to normal activities while minimizing the impact on the number of deaths and avoiding peaks in hospitalization that would saturate healthcare facilities. The strategies that we are investigating are the following:

- **Hard exit**: In this strategy, all mobility activities are resumed to normal on May 11, 2020.
- **Progressive exit**: Mobility activities are gradually restored, with an increase of 15% of the activity every 2 weeks until the pre-lockdown activity level is reached.
- **Cyclic exit**: Every two weeks, activity is resumed to normal then brought back to lock down situation. The process is repeated for 4 cycles, thus ending on 03/08/2020.
- **Status Quo**: The current situation (as of April 30th) is maintained for the entirety of the period.

Figure 8.6 shows the evolution of R_t values for Luxembourg, Italy and Japan. As expected we see no evolution from the initial value in the no exit case, a cyclic fluctuation in the case of a cyclic exit, a soft increase of the R_t values when applying a progressive exit and finally, a brutal jump in the case of a hard exit. R_t reaches a plateau typically quite rapidly after strategies are applied. The plateau depends on the mobility condition, therefore, we see two plateaus in the results, one with all activities remaining closed (no exit), and one where all the other strategies



(a) Luxembourg (b) Italy (c) Japan

Figure 8.6: Evolution of the R_t values for the four exit strategies modelled, *i.e* a hard exit, a progressive exit, a cyclic exit and status quo for Luxembourg, Italy and Japan.

reach the same plateau when the mobility levels are restored to their pre-lockdown baseline.

We evaluate these strategies for the three countries Luxembourg, Italy and Japan and obtain the results depicted by Table 8.4. We choose those three countries because they present difference with respect to their demography, mitigation strategies and number of deaths attributed to COVID-19. Furthermore, they are amongst the few to provide reliable data about hospitalization capacities, hence allowing us to incorporate this information when looking for optimal policy schedules.

We compute for each strategy the Area Under Curve (AUC) of each mobility metric and provide the mean across all mobility values, and compare these hand-crafted strategies with the ones found by our Genetic Algorithm search. The search is run on 100 generations with a size of population of 100 and the hard constraint that critical hospitalizations should not exceed the country's ICU capacity (2,054 for Italy, 1,822 for Japan and 42 for Luxembourg, [MH20a]). All strategies are evaluated between April 30th and September 30th.

We report 2 metrics, the total deaths on September 30th and the mean Area Under Curve across all the mobility features over the 5 months. The later reflects an economic objective that we need to maximize while the former is a healthcare objective. We state in the table three strategies found on the pareto. S1 is the pareto solution with the lowest death toll, S3 is the strategy with the highest mobility activity (and hence highest death toll) and S2 is the median death toll.

Our study shows that progressive lift strategies yield a similar economic footprint as 2-weeks cyclic strategies with fewer casualties (7% fewer deaths for Italy and 10 times less for Luxembourg).

RQ2 Answer: Our approach allows to see drastic changes based on different exit strategies. The progressive strategy offers in our experiments a better outcome than a hard or a cyclic strategy.

The search-based strategies (S1, S2, S3) perform better than manual strategies on the death metric for Italy and Japan, and for Luxembourg. For Luxembourg, S1 performs as good as the progressive strategy. S1 in particular performs on a par with the Progressive lift strategy.

Overall, our results show that the search for exit strategies can be guided and restricted to the policy-makers constraints (i.e. hospital capacity) and yields actionable strategies within constrained computation time. We ran the experiments 100 times (x100 generations each) and the hypervolume of the pareto-solutions converges within 80 generations.

RQ3 Answer: The search algorithm yields better strategies than the popular ones both in term of impact on the activity and number of deaths while ensuring that the healthcare facilities are not overwhelmed.

8.5 Limitations & future work

8.6 Conclusion

In this paper, we studied *DN-SEIR*, a data-driven approach to evaluate the effective reproduction number of the COVID-19 epidemic. In particular, we considered both manual and search-based mitigation strategies, with the aim to help decision-makers in the evaluation and selection of exit strategies. To this end, we evaluated the state-of-the-art compartment model (i.e. SEIR) and shew that our approach yields predictions closer to the ground truth. We also demonstrated that learning can transfer across different countries and a simple FFNN provides accurate and interpretable predictions. Finally, we proposed a search-based approach to evaluate and find optimal strategies that satisfy the constraints of the health facilities and achieve a quick economic recovery with limited casualties.

Our approach paves the ways to automated strategy simulation and search and provides a simple, yet, powerful tool for policy makers to tailor exit strategies to their context and priorities. We can go further than our approach with better feature engineering or neural architecture search (with CNN or RNN). We can also extend the data-driven prediction of hyper-parameters not only to the effective reproduction number but also to all the epidemiological parameters like hospitalization rate. This would require having access to accurate hospitalization data across a large pool of countries and can be achieved in the close future as more countries are sharing such data. Finally, we could extend our technique to a more-grained approach that takes into account age-specific or location-specific epidemiological models.

Country	Strategy	Mobility AUC	Deaths
Luxembourg	Status Quo	-10,721.6	108
	Hard	-91.9	2,763
	Progressive	-2,774.43	114
	Cyclic	-2,487.02	2,002
	Pareto-S1	-2,381.45	165
	Pareto-S2	-2,370.7	635
	Pareto-S3	-2,289.7	697
Italy	Status Quo	-6,006.13	32,015
	Hard	-57.53	37,377
	Progressive	-4,124.93	31,987
	Cyclic	-3,689.13	34,427
	Pareto-S1	-8,412.125	29,449
	Pareto-S2	-7,570.0	29,450
	Pareto-S3	-7,275.38	29,452
Japan	Status Quo	-3,431.7	709
	Hard	14.52	710
	Progressive	- 1,005.3	708
	Cyclic	-896.03	709
	Pareto-S1	-2,106.21	654
	Pareto-S2	-1,170.33	660
	Pareto-S3	-1,106.33	671

Table 8.4: Exit strategies comparison. Higher AUC and lower deaths are better.

9

Conclusion

This chapter presents the overall conclusion of the dissertation and proposes potential research directions.

Contents

9.1 Summary of contributions

In this dissertation, we present studies, techniques, and algorithms that improve the robustness of ML systems to test inputs that exhibit feature shifts compared to the training set. We cover shifts caused by malicious third parties (i.e., adversarial attacks), shifts caused by confounding factors (i.e., unknown/uncontrolled parameters in the data collection process), and shifts caused by time drifts and changes in the observed phenomena. Indeed, each phenomenon can be best studied in a different use case and requires a set of techniques and tools best suited to answer its specific challenges and constraints. Thereupon, this dissertation brings the following contributions: (1) an empirical study that demonstrates how domain knowledge is key to the robustness of complex ML systems; (2) a set of novel approaches to leverage domain knowledge and augment ML models through multi-objective learning to improve the robustness of the ML system; (3) a large-scale study to demonstrate that multi-objective learning using confounding attributes also improves the generalization of ML models.

The first contribution aims to shed light on how robust ML models are used in critical systems with scarce high-quality data. We first cover the use-case of a financial ML system, the credit scoring system deployed by BGL BNP Paribas to handle overdraft authorizations. Our analysis demonstrates that the feature engineering with hand-made rules and the domain knowledge involved in the data preparation makes it impossible for an adversary to attack their system with reasonable resources. However, once the system's constraints are disclosed, the attacker can successfully generate adversaries to cause significant financial damage to our industrial partner. Therefore, we design new attacks and defense mechanisms to close the remaining gaps and loopholes.

Similarly, we demonstrate in the second use case that multi-task models used in robot navigation and autonomous vehicles are harder to attack because of the complexity of the fusing mechanism. Indeed, the shared layers in multi-task models enforce domain constraints across the learned tasks. Therefore, we design attacks aware of these natural protections and propose new defense mechanisms to mitigate these attacks.

For the second contribution, our objectives were to assess whether the multi-task mechanism previously uncovered can be leveraged on any computer vision ML model to improve its robustness. With this in mind, we design a new framework, **Task Augmentation**, to augment any existing computer-vision model with ad-hoc tasks to improve its robustness. Indeed, we demonstrate that one can extend the original model with domain-specific tasks (extracted from the images' meta-data) or self-supervised tasks (generated on the go with image processing techniques) to enforce semantic constraints and improve the robustness of the models.

First, we evaluate this approach against adversarial attacks for chest radio-

graph pathology classification. We demonstrate that our approach improves the robustness of the models against strong attacks and outperforms most recent data-augmentation techniques, especially for pathologies with scarce data. Next, we show that *Task Augmentation* also improves the generalization of models to unseen datasets. We evaluate our approach by training and testing chest radiographs from different hospitals and countries and demonstrate that task augmentation leads to significant generalization improvements. Moreover, it is competitive with computationally expensive approaches like ensemble learning and unlabeled data augmentations. Finally, we demonstrate that one can improve generalization performance by leveraging confounding factors like age and gender.

Our third and last contribution spawns from the intuition that multi-objective learning leveraging cofounding attributes can also improve the generalization of ML forecasting models. Taking part in the Luxembourg Covid19 task force, we designed a fully integrated tool to forecast the pandemic and search for optimal mitigation strategies that balance the human and economic losses.

9.2 Broader Impact

Although our work focused on concrete and industrial cases, our contributions pioneered the research in two fields of robust machine learning.

- **robustness of constrained neural networks.** In recent years, there has been a growing interest in using expert-designed domain-knowledge to create neural models that (1) perform better and (2) learn from fewer data. Our work is the first to explore how adding knowledge to the models can improve their robustness to distribution shift and adversarial perturbations. Our approaches require less data and training computation to achieve equivalent performances than data-hungry approaches. While recent SoTA data-augmentations have outperformed our techniques, we expect our work to pave the way to further research in the area. For example, a follow-up line of research would be to investigate the combination of domain-knowledge and data-augmentation techniques.
- **robustness of physics-informed neural networks.** Physics-informed neural networks are neural networks trained to solve supervised learning tasks while complying with any physical laws described by general nonlinear partial differential equations. In our Covid19 study, we proposed a take on this family of models by integrating an epidemiological differential equation model with our neural network. Our work is among the first cornerstones for augmenting neural networks with domain knowledge using differential equations and achieving higher robustness and reliability. Consequently, we advocate for more research on exploiting structured prior information in the form of differential equation models to build data-efficient and physics-

informed models. Especially, many learning processes can be found in nature in the form of partial differential equations. It is only fair to assume that a better understanding of their dynamics and incorporation within the neural networks will lead to a giant leap for ML research.

10

Appendices

This chapter includes the appendices related to each individual chapter.

Contents

10.1 Search-Based Adversarial Testing and Improvement of Constrained Credit Scoring Systems.

This paper does not have any reference any appendix.

10.2 Adversarial Robustness in Multi-Task Learning: Promises and Illusions.

10.2.1 Appendix A: Proofs for the Theoretical Analysis

Definition 8. *Let $\mathscr{M}$ be a multi-task model. $\mathscr{T}' \subseteq \mathscr{T}$ a subset of its tasks and $\mathcal{L}'_{\mathscr{T}}$ the joint loss of tasks in $\mathscr{T}'$. Then, we call $\mathbb{E}_x[\delta\mathcal{L}(\mathscr{T}', \epsilon)]$ the* adversarial vulnerability *of $\mathscr{M}$ on $\mathscr{T}'$ to an ϵ-sized $\|.\|_p$-attack.*

And we define it as the average increase of $\mathcal{L}_{\mathscr{T}'}$ after attack over the whole dataset, i.e.:

$$\mathbb{E}_x[\delta\mathcal{L}(\mathscr{T}', \epsilon)] = \mathbb{E}_x\left[\max_{\|\delta\|_p \leq \epsilon} \mid \mathcal{L}_{\mathscr{T}'}(x+\delta, \bar{y}) - \mathcal{L}_{\mathscr{T}'}(x, \bar{y}) \mid\right]$$

Lemma 1. *Under an ϵ-sized $\|.\|_p$-attack, the adversarial vulnerability of a multi-task model can be approximated through the first-order Taylor expansion, that is:*

$$\mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}')] \approx \epsilon \cdot \mathbb{E}_x\big[\mid\mid \partial_x\mathcal{L}'(x, \bar{y}) \mid\mid_q\big] \tag{10.1}$$

Proof. **10.2.2 A.1**

From definition 1, we have:

$$\mathbb{E}_x[\delta\mathcal{L}(\mathscr{T}', \epsilon)] \quad = \quad \mathbb{E}_x\left[\max_{\|\delta\|_p \leq \epsilon} \mid \mathcal{L}_{\mathscr{T}'}(x+\delta, \bar{y}) - \mathcal{L}_{\mathscr{T}'}(x, \bar{y}) \mid\right]$$

Given the perturbation δ is minimal, we can approximate $\delta\mathcal{L}$ with a Taylor expansion up to a second order:

$$\mathbb{E}_x[\delta\mathcal{L}(\mathscr{T}', \epsilon)] \quad \approx \quad \mathbb{E}_x\left[\max_{\|\delta\|_p \leq \epsilon} \mid \delta \cdot \partial_x\mathcal{L}'(x, \bar{y}) + \frac{\delta^2}{2} \cdot \partial_x^2\mathcal{L}'(x, \bar{y}) \mid\right]$$

The noise δ is optimally adjusted to the coordinates of $\partial_x\mathcal{L}'$ within an ϵ-constraint. By the definition of the dual-norm, we get:

$$\mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}')] \approx \mathbb{E}_x\left[\mid\mid \epsilon \cdot \partial_x\mathcal{L}'(x, \bar{y}) + \frac{\epsilon^2}{2} \cdot \partial_x^2\mathcal{L}'(x, \bar{y}) \mid\mid_q\right] \tag{10.2}$$

where q is the dual norm of p and $\frac{1}{p} + \frac{1}{q} = 1$ and $1 \leq p \leq \infty$.

We obtain Lemma 2 by restricting the Taylor expansion to the first-order.

□

Theorem 3. *Consider a multi-task model $\mathscr{M}$ where an attacker targets $\mathscr{T} = \{t_1, ..., t_M\}$ tasks uniformly weighted, with an ϵ-sized $\|.\|_p$-attack. If the model is converged, and the gradient for each task is i.i.d. with zero mean and the tasks are correlated, the adversarial vulnerability of the model can be approximated as*

$$\mathbb{E}_x[\delta\mathcal{L}'] \approx K \cdot \sqrt{\frac{1 + \frac{2}{M}\sum_{i=1}^{M}\sum_{j=1}^{i-1}\frac{\text{Cov}\left(\mathbf{r}_i, \mathbf{r}_j\right)}{\text{Cov}(\mathbf{r}_i, \mathbf{r}_i)}}{M}} \tag{10.3}$$

where K is a constant dependant of ϵ and the attacked tasks and $\mathbf{r}_i = \partial_x\mathcal{L}(x, y_i)$ the gradient of the task i and $\text{Cov}\left(\mathbf{r}_i, \mathbf{r}_j\right)$ the covariance between the two gradients $\mathbf{r}_i, \mathbf{r}_j$.

Proof. **10.2.3 A.2**

let $\mathbf{r}_i = \partial_x\mathcal{L}(x, y_i)$ the gradient of the task i, with a weight $w_i = \frac{1}{M}$ such as the joint gradient of $\mathscr{M}$ is defined as $\partial_x\mathcal{L}(x, \bar{y}) = \sum_{i=1}^{M} w_i\mathbf{r}_i$. let $p = q = 2$

We have:

$$\begin{aligned}
\mathbb{E}_x[|| \epsilon \cdot \partial_x\mathcal{L}'(x, \bar{y}) ||_2^2] &= \mathbb{E}_x\left[|| \sum_{j=1}^{M}\frac{\epsilon}{M} \cdot r_i ||_2^2\right] \\
= \frac{\epsilon^2}{M^2}\mathbb{E}_x\left[\sum_{i=1}^{M}|| \mathbf{r}_i ||_2^2 + 2\sum_{i=1}^{M}\sum_{j=1}^{i-1}|| \mathbf{r}_i ||_2|| \mathbf{r}_j ||_2\right] \\
&= \frac{\epsilon^2}{M^2}\left(\sum_{i=1}^{M}\mathbb{E}_x[\mathbf{r}_i^2] + 2\sum_{i=1}^{M}\sum_{j=1}^{i-1}\mathbb{E}_x[\mathbf{r}_i\mathbf{r}_j]\right)
\end{aligned} \tag{10.4}$$

We know:

$$\text{Cov}\left(\mathbf{r}_i, \mathbf{r}_j\right) = \mathbb{E}_x\left[\mathbf{r}_i\mathbf{r}_j\right] - \mathbb{E}_x\left[\mathbf{r}_i\right]\mathbb{E}_x\left[\mathbf{r}_j\right] \tag{10.5}$$

According to the assumptions, the gradient of each task is i.i.d with zero means: $\mathbb{E}_x\left[\mathbf{r}_i\right] = 0$ Then $\text{Cov}\left(\mathbf{r}_i, \mathbf{r}_j\right) = \mathbb{E}_x\left[\mathbf{r}_i\mathbf{r}_j\right]$ and $\sigma_i^2 = \text{Cov}\left(\mathbf{r}_i, \mathbf{r}_i\right) = \mathbb{E}_x\left[\mathbf{r}_i^2\right]$.

$$\begin{aligned}
\mathbb{E}_x[|| \epsilon \cdot \partial_x\mathcal{L}'(x, \bar{y}) ||_2^2] &= \frac{\epsilon^2}{M^2}\sum_{i=1}^{M}\left(\sigma_i^2 + 2\sum_{j=1}^{i-1}\text{Cov}\left(\mathbf{r}_i, \mathbf{r}_i\right)\right) \\
&\propto \frac{1}{M}\left(1 + 2\sum_{i=1}^{M}\sum_{j=1}^{i-1}\frac{\text{Cov}\left(\mathbf{r}_i, \mathbf{r}_i\right)}{M\sigma_i^2}\right) \\
\mathbb{E}_x[|| \epsilon \cdot \partial_x\mathcal{L}'(x, \bar{y}) ||_2] &\propto \sqrt{\frac{\left(1 + 2\sum_{i=1}^{M}\sum_{j=1}^{i-1}\frac{\text{Cov}(\mathbf{r}_i, \mathbf{r}_i)}{M\sigma_i^2}\right)}{M}}
\end{aligned} \tag{10.6}$$

Using the first order adversarial vulnerability (Lemma 2), we then have:

$$\mathbb{E}_x[\delta\mathcal{L}'] \approx K \cdot \sqrt{\frac{1 + \frac{2}{M}\sum_{i=1}^{M}\sum_{j=1}^{i-1}\frac{\mathrm{Cov}(\mathbf{r}_i,\mathbf{r}_j)}{\mathrm{Cov}(\mathbf{r}_i,\mathbf{r}_i)}}{M}} \tag{10.7}$$

with K a constant dependant of ϵ and the attacked tasks.

□

Definition 9. *Let $\mathscr{M}$ be a multi-task model with $\mathscr{T}_M = \{t_1, ..., t_M\}$ tasks, an input x, $\bar{y} = (y_1, ..., y_M)$ its corresponding ground-truth. We denote the set of attacked tasks $\mathscr{T}_N$ and $\mathscr{T}_{N+1}$, two subsets of the model's tasks $\mathscr{T}$ such as $\mathscr{T}_{N+1} = \mathscr{T}_N \cup \{t_{N+1}\}$ and $N + 1 \leq M$, and let $\mathcal{L}'$ be the joint task loss of attacked tasks.*

We call marginal adversarial vulnerability *of the model to an $\mathscr{T}', \epsilon$-sized $\|.\|_p$-attack the difference between the adversarial vulnerability over the task set $\mathscr{T}_{N+1}$ and the adversarial vulnerability over the task set $\mathscr{T}_N$.*

$$\Delta_N\mathbb{E}_x[\delta\mathcal{L}'] = \mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_{N+1})] - \mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_N)]$$

Lemma 2. *Under an ϵ-sized $\|.\|_p$-attack, the marginal adversarial vulnerability of a multi-task model can be approximated through the first-order Taylor expansion, that is:*

$$\Delta_N\mathbb{E}_x[\delta\mathcal{L}'] \approx \widetilde{\Delta_N\mathbb{E}_x[\delta\mathcal{L}']} = \epsilon \cdot \left(\mathbb{E}_x\left[\,||\ \partial_x\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_{N+1})\ ||_q\right] - \mathbb{E}_x\left[\,||\ \partial_x\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_N)\ ||_q\right]\right)$$

Proof. **10.2.4 A.3**

From Definition 4, we have:

$$\Delta_N\mathbb{E}_x[\delta\mathcal{L}'] = \mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_{N+1})] - \mathbb{E}_x[\delta\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_N)]$$

using the lemma 2 at the first order expansion on each term of the right side, we get:

$$\Delta_N\mathbb{E}_x[\delta\mathcal{L}'] \approx \epsilon \cdot \mathbb{E}_x\left[\,||\ \partial_x\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_{N+1})\ ||_q\right] - \epsilon \cdot \mathbb{E}_x\left[\,||\ \partial_x\mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_N)\ ||_q\right]$$

□

Lemma 3. *For a given multi-task model $\mathscr{M}$, let $\mathbf{r}_i = \partial_x\mathcal{L}(x, y_i)$ the gradient of the task i, with a weight w_i such as the joint gradient of $\mathscr{M}$ is defined as $\partial_x\mathcal{L}(x, \bar{y}) = \sum_{i=1}^{M} w_i\mathbf{r}_i$. Let $\|.\|_q$ be a norm and p an integer. We have:*

$$\mathbb{E}_x[||\sum_{i=1}^{M} w_i\mathbf{r}_i\ ||_q^p] \leq \sum_{i=1}^{M} w_i^p\mathbb{E}_x\left[||\ \mathbf{r}_i\ ||_q^p\right] \tag{10.8}$$

Proof. **10.2.5 A.4**

$$\begin{aligned}\mathbb{E}_x[|| \sum_{i=1}^{M} w_i \mathbf{r}_i ||_q^p] &\leq \mathbb{E}_x \left[\sum_{i=1}^{M} || w_i \mathbf{r}_i ||_q^p \right] \\ &\leq \sum_{i=1}^{M} w_i^p \mathbb{E}_x \left[|| \mathbf{r}_i ||_q^p \right]\end{aligned} \tag{10.9}$$

□

This lemma provides an upper-bound of the average norm of the gradients that we use to evaluate the upper bounds of the adversarial vulnerability in the following theorem:

Theorem 4. *For a given multi-task model $\mathscr{M}$, let $\mathbf{r}_i = \partial_x \mathcal{L}(x, y_i)$ the gradient of the task i, with a weight w_i and zero mean such as the joint gradient of $\mathscr{M}$ is defined as $\partial_x \mathcal{L}(x, \bar{y}) = \sum_{i=1}^{M} w_i \mathbf{r}_i$. The first order marginal vulnerability is bounded as follow:*

$$\begin{aligned}\widetilde{\Delta_N \mathbb{E}_x[\delta \mathcal{L}']} \leq \epsilon \cdot ((N+1) \cdot w_{N+1} \mathbb{E}_x[|| \mathbf{r}_{N+1} ||] + \\ N \cdot \max_{i<N+1} w_i \mathbb{E}_x[|| \mathbf{r}_i ||])\end{aligned}$$

Proof. **10.2.6 A.5**

Using Lemma 5, we have:

$$\begin{aligned}\Delta_N \mathbb{E}_x[\delta \mathcal{L}'] &\approx \epsilon \cdot \mathbb{E}_x[|| \partial_x \mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_{N+1}) ||_q] - \epsilon \cdot \mathbb{E}_x[|| \partial_x \mathcal{L}'(x, \bar{y}, \epsilon, \mathscr{T}_N) ||_q] \\ &\leq \epsilon \left(| \mathbb{E}_x[|| \partial_x \mathcal{L}'(\mathscr{T}_{N+1}) ||_q] | + | \mathbb{E}_x[|| \partial_x \mathcal{L}'(\mathscr{T}_N) ||_q] | \right)\end{aligned} \tag{10.10}$$

We use lemma 6 with p=1 and N+1:

$$\begin{aligned}\mathbb{E}_x[|| \sum_{i=1}^{N+1} w_i \mathbf{r}_i ||_q] &\leq (N+1) \sum_{i=1}^{N+1} w_i \mathbb{E}_x \left[|| \mathbf{r}_i ||_q \right] \\ &\leq (N+1) \left(\sum_{i=1}^{N} w_i \mathbb{E}_x \left[|| \mathbf{r}_i ||_q \right] + w_{N+1} \mathbb{E}_x \left[|| \mathbf{r}_{N+1} ||_q \right] \right) \\ \leq N \sum_{i=1}^{N} w_i \mathbb{E}_x \left[|| \mathbf{r}_i ||_q \right] &+ \sum_{i=1}^{N} w_i \mathbb{E}_x \left[|| \mathbf{r}_i ||_q \right] + (N+1) \cdot w_{N+1} \mathbb{E}_x \left[|| \mathbf{r}_{N+1} ||_q \right]\end{aligned} \tag{10.11}$$

We use similarly lemma 6 with p=1 and N:

$$\mathbb{E}_x[|| \sum_{i=1}^{N} w_i \mathbf{r}_i \; ||_q] \leq N \sum_{i=1}^{N} w_i \mathbb{E}_x \left[|| \; \mathbf{r}_i \; ||_q\right] \tag{10.12}$$

We inject (11) and (12) in (10) and we have:

$$\Delta_N \mathbb{E}_x[\delta \mathcal{L}'] \leq \epsilon \left((N+1) \cdot w_{N+1} \mathbb{E}_x[|| \; \mathbf{r}_{N+1} \; ||] + \sum_{i=1}^{N} w_i \mathbb{E}_x[|| \; \mathbf{r}_i \; ||] \right)$$

$$\Delta_N \mathbb{E}_x[\delta \mathcal{L}'] \leq \epsilon \left((N+1) \cdot w_{N+1} \mathbb{E}_x[|| \; \mathbf{r}_{N+1} \; ||] + N \max_{i<N+1} w_i \mathbb{E}_x[|| \; \mathbf{r}_i \; ||] \right)$$

□

10.2.7 Appendix B: Experimental Settings

General training We use the same learning rate schedule for all the models: SGD with learning rate 0.01 and momentum 0.99. We decrease the learning rate at 100 epoch by 10 times, then successively at epoch=120 and epoch=140, we decrease again by 10 times. We train all the models for 150 epochs.

We train on 80% of the rooms (9464 images from 1500 different rooms) and test on the remaining 20%.

Experimental Settings We train different combinations of encoders and task decoders: Resnet18, Resnet50, W-Resnet50, Resnet152 and Xception. This allows us to check that our hypothesis of the limited impact of multi-task learning to generalize across different families of architectures and sizes. Table 10.1 lists our different settings. We evaluate the cost of the models as number of FLOPS (Floating Points Operations) required for one image inference, while the size is the number of weights of the model. Each task is handled by a specific decoder. The decoders are 8 layers (Convolution & Dense).

Setting	Encoder	Weighted	Tasks	#Models	# Epochs	Size	Cost
S1	Resnet18	Uniform	(s,d,D,n,k,K,e,E,p,r,A)	121	150	14.19M	6.09B
S2	Resnet18	Uniform	(s,d,D,n,k,K,e,E,p,r,A)	121	50	14.19M	6.09B
S3	Resnet18	Uniform	(s,d,D,n,k,K,e,E,p,r,A)	121	100	14.19M	6.09B
S4	Resnet18	Optimal	(s,d,D,n,E)	25	150	14.19M	6.09B
S5	Resnet50	Uniform	(s,d,D,n,E)	25	150	29.66M	9.90B
S6	Xception	Uniform	(s,d,D,n,E)	25	150	4.33M	3.64B
S7	Resnet152	Uniform	(s,d,D,n,E)	25	150	64.30M	19.64B
S8	Wide-Resnet50	Uniform	(s,d,D,n,E)	25	150	72.99M	19.45B

Table 10.1: The experimental settings we evaluated.

List of weights for the weighted models For the weighted setting (S4), we use these weights. For each of the main tasks, we use 1 as a weight and the following for the auxiliary tasks weights:

- **Semantic segmentation (s)**: 0.01 for sd combination, 0.1 otherwise.
- **Z-Depth (d)**: 0.01 for dn combination, 0.1 otherwise.
- **Normal (n)**: 0.01 for nd combination, 0.1 otherwise.
- **Euclidian Depth (D)**: 0.1 for Ds combination, 0.01 otherwise.
- **Edge detection (E)**: 0.1 for all combinations.

10.2.8 Appendix C: Detailed evaluation of the settings and tasks

Relative task robustness of architectures

We provide in Tables 10.2 to 10.6 the clean performance of each task combination. We also provide the relative task vulnerability against single-task and multi-task attacks.

While the multi-task models are not reliably more robust than single-task models across all architectures, we see that robust combinations are similar across different models.

Auxiliary →		s	d	D	n	E
Clean	s	50.50	49.20	47.28	47.02	48.63
	d	101.08	97.50	98.48	93.94	100.53
	D	96.71	91.30	92.36	91.23	87.08
	n (e^{-2})	71.89	57.66	58.59	54.56	57.41
	E (e^{-2})	16.43	10.30	10.49	11.68	8.78
Single	s	0.81	0.84	0.97	0.96	0.93
	d	7.73	7.87	7.46	10.67	10.55
	D	7.78	8.92	8.44	10.95	11.78
	n	9.43	13.49	11.61	15.02	13.40
	E	16.32	26.85	25.43	26.62	31.37
Multi	s	0.81	0.81	0.94	0.92	0.90
	d	2.90	7.87	7.71	5.96	4.36
	D	3.01	8.91	8.44	5.83	5.61
	n	5.97	13.44	11.82	15.02	11.43
	E	10.02	24.95	24.15	18.62	31.37

Table 10.2: Relative task vulnerability (lower is better) for the **Resnet18** models. Each row refers to the main task evaluated and the column to the auxiliary task. In the top half (Clean), we provide the clean performance (1-mIoU for s, MSE for the rest), in the middle (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

Auxiliary →		s	d	D	n	E
Clean	s	51.63	49.50	49.66	47.48	46.90
	d	106.57	98.28	95.92	95.35	94.38
	D	99.65	89.81	90.58	89.17	91.26
	n (e^{-2})	73.46	46.47	44.97	49.91	48.34
	E (e^{-2})	15.30	7.48	8.93	9.04	6.99
Single	s	0.82	0.90	0.90	0.97	0.99
	d	7.89	6.06	7.73	11.09	10.59
	D	3.13	2.19	3.02	4.46	1.76
	n	10.36	17.10	18.28	16.26	16.52
	E	22.67	29.75	22.11	31.07	28.41
Multi	s	0.82	0.88	0.89	0.94	0.96
	d	3.79	6.06	7.78	6.93	4.78
	D	1.57	2.16	3.02	2.27	0.89
	n	7.46	17.13	18.21	16.26	14.03
	E	14.00	28.26	21.95	22.54	28.41
-						

Table 10.3: Relative task vulnerability (lower is better) for the **Resnet50** models. Each row refers to the main task evaluated and the column to the auxiliary task. In the top half (Clean), we provide the clean performance (1-mIoU for s, MSE for the rest), in the middle (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

Auxiliary →		s	d	D	n	E
Clean	s	42.91	42.14	43.19	44.35	44.17
	d	109.27	88.66	90.67	92.55	93.21
	D	103.55	98.43	85.70	93.51	88.58
	n (e^{-2})	5183.49	3932.54	4065.30	3803.32	4178.78
	E (e^{-2})	1817.37	140453.97	142222.13	2028.51	543.58
Single	s	1.21	1.26	1.20	1.11	1.14
	d	7.87	14.39	14.47	28.00	25.66
	D	8.18	19.31	12.88	7.33	23.42
	n	14.09	20.08	17.40	22.03	19.23
	E	18.03	0.15	0.14	22.44	193.38
Multi	s	1.21	1.23	1.16	1.07	1.09
	d	3.69	14.39	14.38	15.62	9.43
	D	3.58	17.66	12.88	3.91	9.18
	n	8.15	19.97	17.49	22.03	16.68
	E	11.38	0.10	0.10	22.75	193.38

Table 10.4: Relative task vulnerability (lower is better) for the **Xception** models. Each row refers to the main task evaluated and the column to the auxiliary task. In the top half (Clean), we provide the clean performance (1-mIoU for s, MSE for the rest), in the middle (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

Auxiliary →		s	d	D	n	E
Clean	s	46.87	54.05	54.78	54.17	56.42
	d	116.91	180.59	403.70	108.81	114.66
	D	127.78	160.12	130.55	105.36	117.93
	n (e^{-2})	85.78	52.34	60.13	54.96	52.68
	E (e^{-2})	20.95	11.74	8.66	10.35	14.14
Single	s	1.01	0.75	0.74	0.74	0.67
	d	9.88	9.95	3.35	12.61	6.71
	D	5.91	9.65	11.65	9.93	12.70
	n	9.14	15.62	13.26	17.94	15.10
	E	11.95	20.23	30.85	25.41	17.81
Multi	s	1.01	0.74	0.73	0.72	0.65
	d	3.76	9.95	3.19	10.64	3.13
	D	2.25	8.54	11.65	5.67	6.49
	n	6.29	15.11	13.07	17.94	13.58
	E	8.16	19.67	32.11	17.73	17.81

Table 10.5: Relative task vulnerability (lower is better) for the **WideResnet50** models. Each row refers to the main task evaluated and the column to the auxiliary task. In the top half (Clean), we provide the clean performance (1-mIoU for s, MSE for the rest), in the middle (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

Auxiliary →		s	d	D	n	E
Clean	s	46.15	45.17	50.48	51.01	50.86
	d	109.27	98.27	107.10	95.34	171.99
	D	103.55	92.58	96.60	93.51	90.10
	n (e^{-2})	110.84	48.20	56.71	59.22	116.94
	E (e^{-2})	18.17	18.20	6.97	20.29	10.52
Single	s	1.03	1.07	0.87	0.82	0.87
	d	7.87	6.87	6.93	9.03	2.65
	D	8.18	7.45	7.19	7.33	8.03
	n	11.10	14.29	13.51	10.87	22.06
	E	18.03	54.91	16.02	22.44	9.41
Multi	s	1.03	1.05	0.85	0.80	0.85
	d	3.69	6.87	7.18	5.60	1.83
	D	3.58	7.30	7.19	3.91	3.44
	n	8.64	14.35	13.43	10.87	21.94
	E	11.38	51.44	15.80	22.75	9.41

Table 10.6: Relative task vulnerability (lower is better) for the **Resnet152 models**. Each row refers to the main task evaluated and the column to the auxiliary task. In the top half (Clean), we provide the clean performance (1-mIoU for s, MSE for the rest), in the middle (Single), we only attack the main task, in the bottom half (Multi), both tasks are attacked.

Performance over all 11 tasks

Figures 10.1-10.11 show the performance of the Resnet18 models after attack (25-steps PGD l_∞ with $\epsilon = 8/255; \alpha = 2/255$).

Across all 11 tasks, the multi-task models are not more robust than their single-task counterparts. Some are, while most are not.

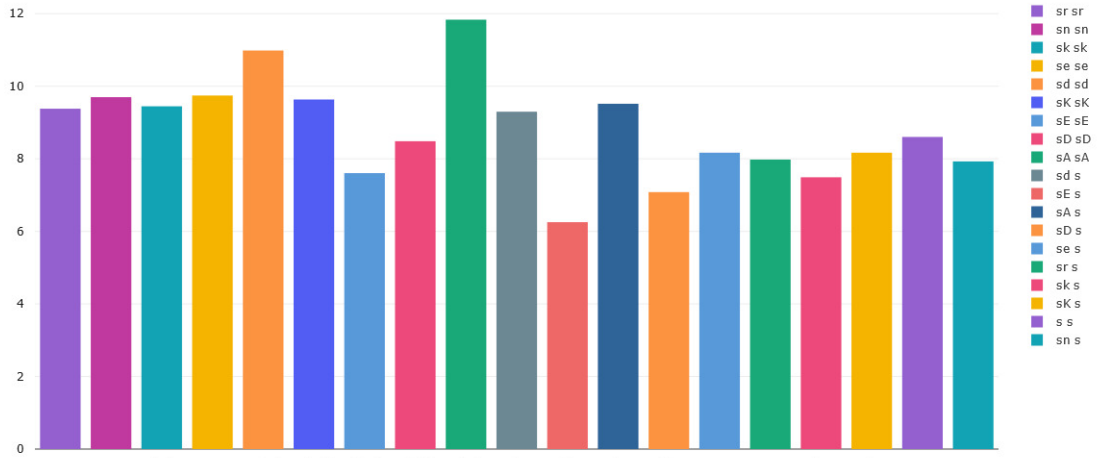


Figure 10.1: mIoU Semantic Segmentation (s) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. For instance “sn s" means model trained on both tasks s and n but only task s attacked. “s s" is the single-task baseline.

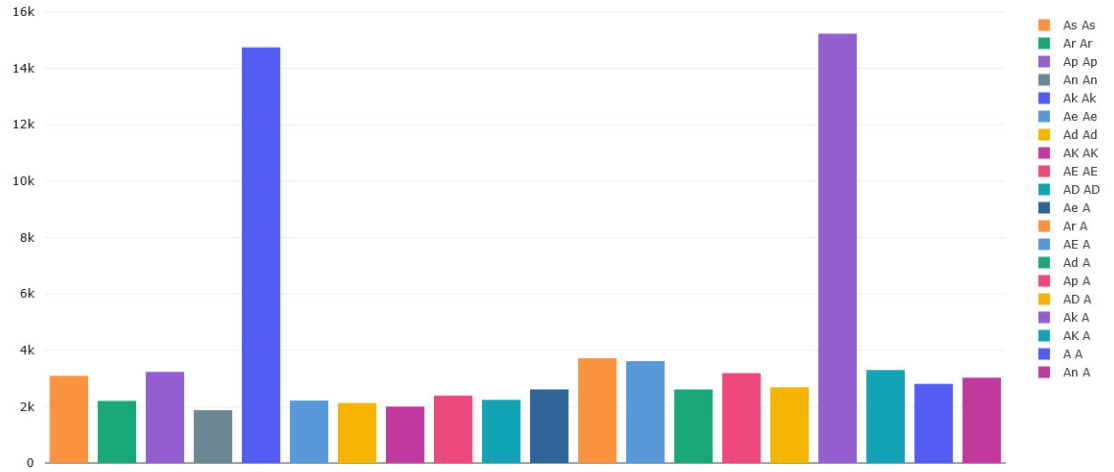


Figure 10.2: MSE of the Auto-encoder task (A) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. “A A" is the single-task baseline.

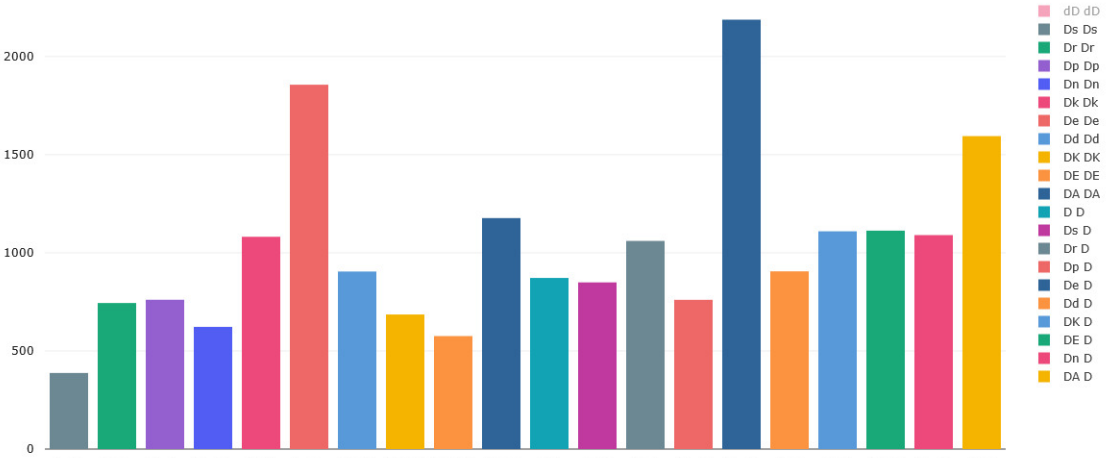


Figure 10.3: MSE of the Euclidian Depth (D) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. “D D" is the single-task baseline.

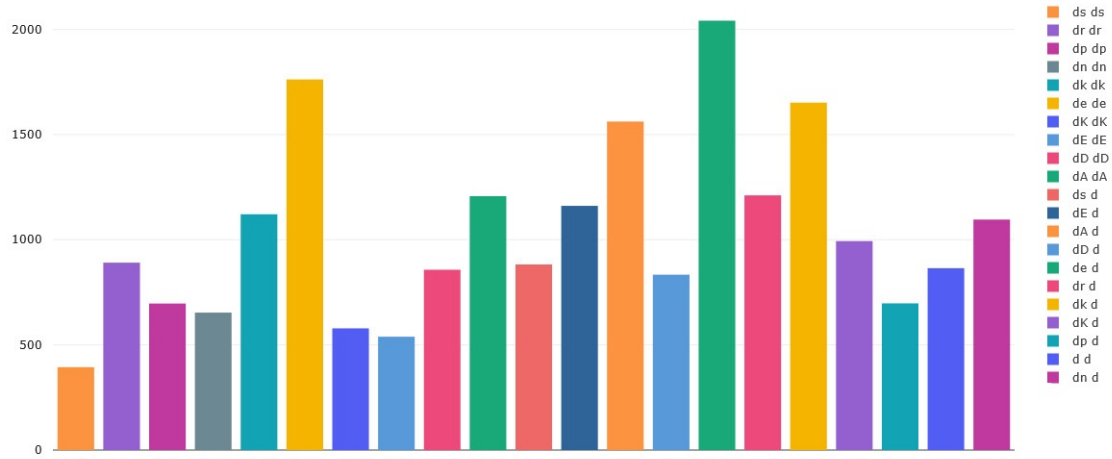


Figure 10.4: MSE of the Z-Depth (d) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. “d d" is the single-task baseline.

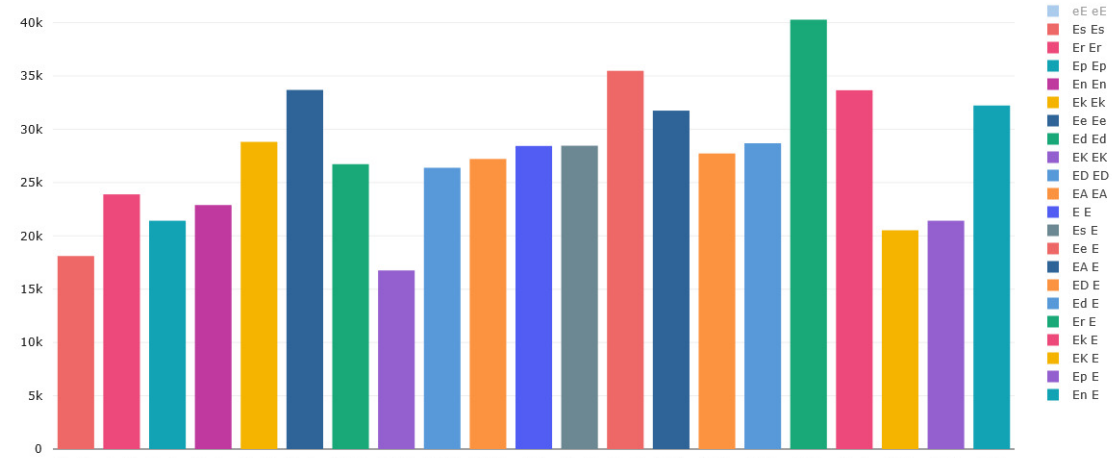


Figure 10.5: MSE of the Edge Occlusion (E) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. “E E" is the single-task baseline.

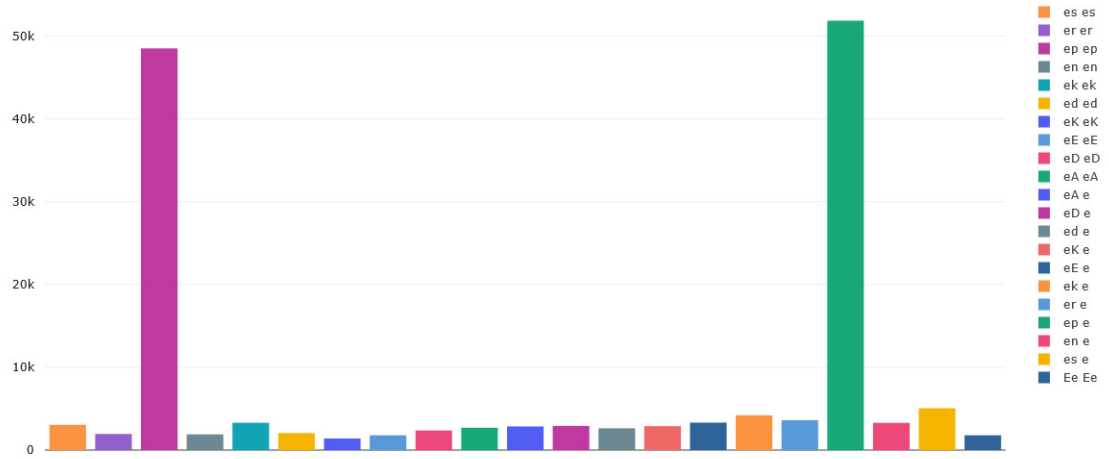


Figure 10.6: MSE of the Edge Texture (e) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. “e e" is the single-task baseline.

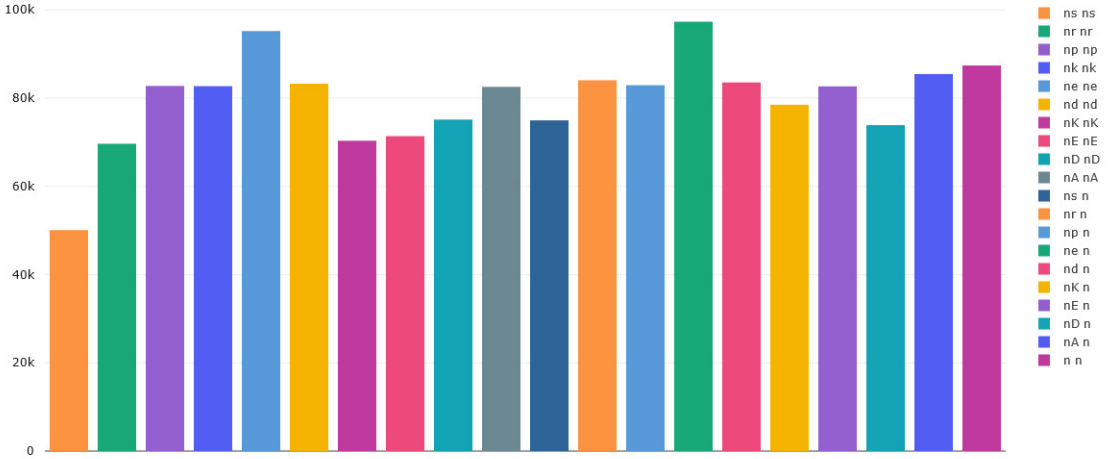


Figure 10.7: MSE of the Edge Normal estimation (n) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. “n n" is the single-task baseline.

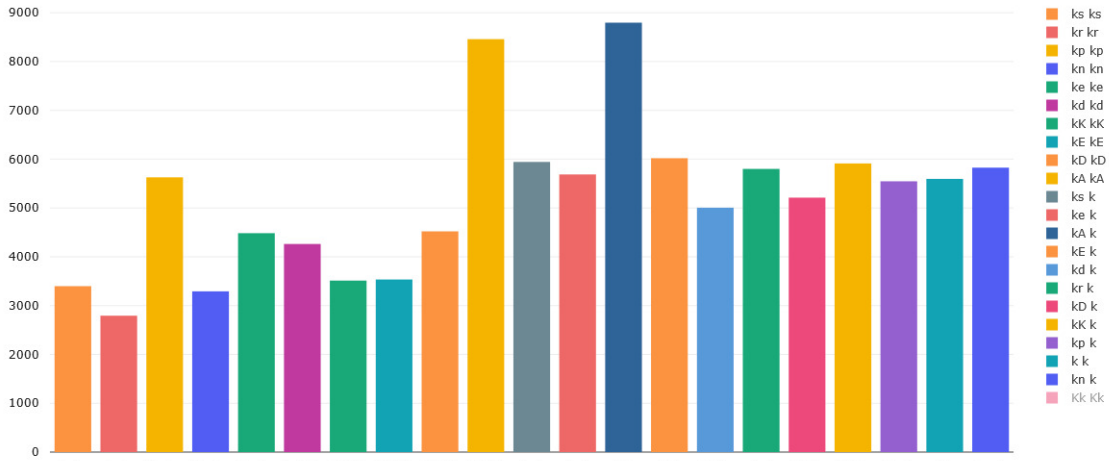


Figure 10.8: MSE of the Keypoints 2d (k) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. “k k" is the single-task baseline.

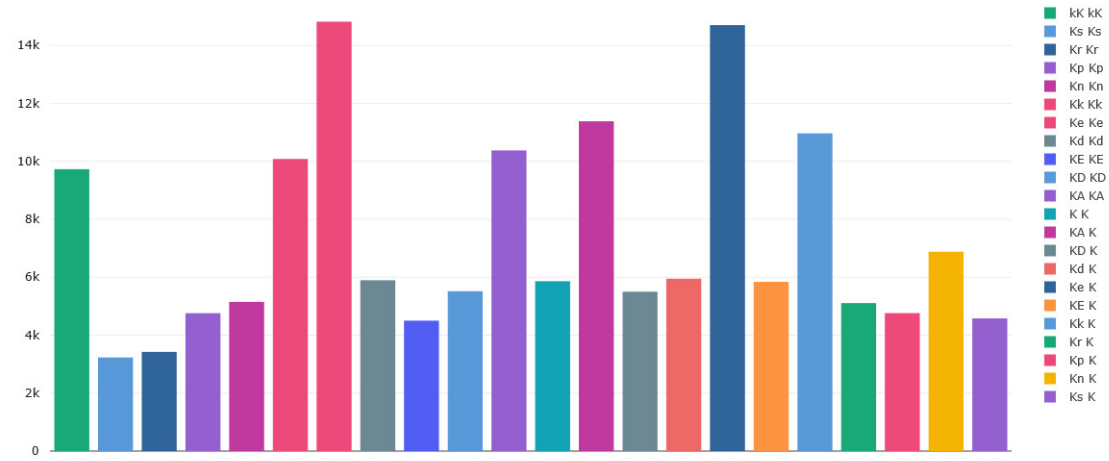


Figure 10.9: MSE of the Keypoints 3d (K) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. “K K" is the single-task baseline.

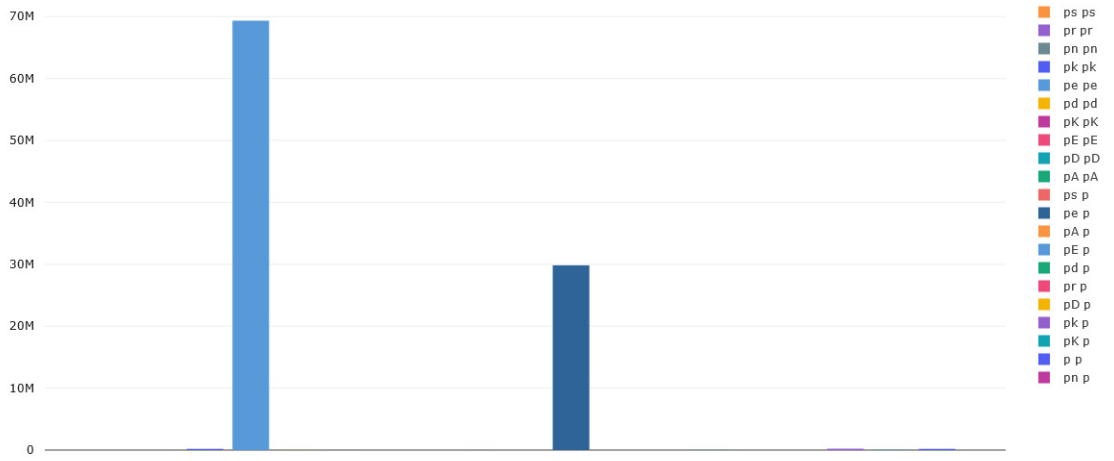


Figure 10.10: MSE of the Principal curvature (p) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. “p p" is the single-task baseline.

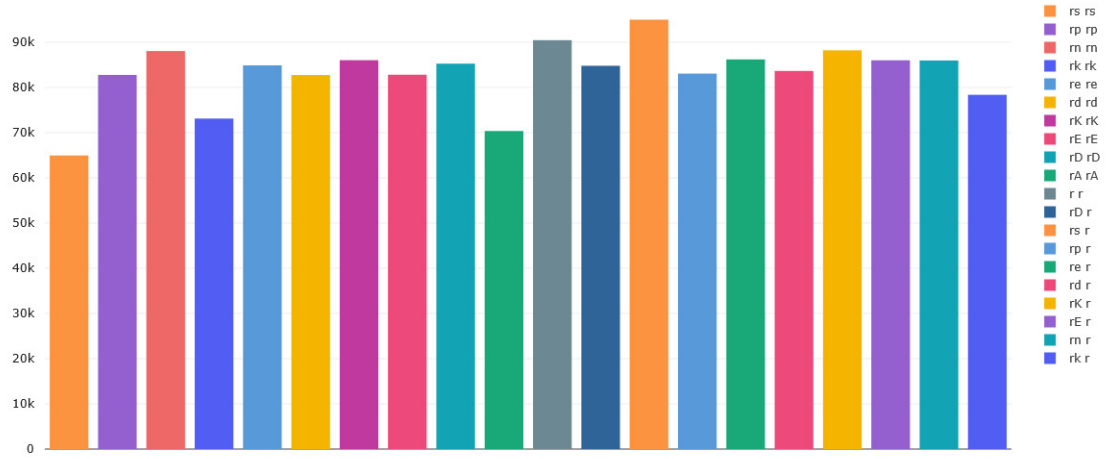


Figure 10.11: MSE of the Reshading (r) after adversarial attack. Legends: first combination of letters are the classes used for training, second combination of letters are tasks attacked. “r r" is the single-task baseline.

Adversarial Vulnerability and number of tasks

In addition to the main paper Figure 1, we evaluate the scenario where the tasks are added successively and the whole model is trained.

Figures 10.12 and 10.13 show how adversarial vulnerability changes when adding additional tasks. When the tasks are not weighted, the additional tasks do not improve the robustness of the models. When the tasks are weighted however, we can see that except when adding vulnerable tasks (s or E in our examples), the models vulnerability tends to decrease when adding supplementary tasks.

These results confirm our main claims that the number of tasks is not the main factor of the vulnerability of multi-task models but how we choose the tasks and how we weigh them.

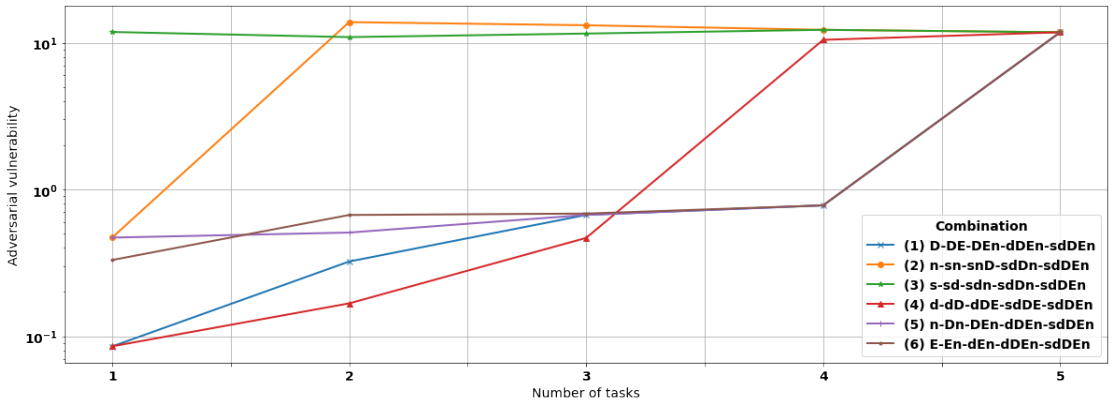


Figure 10.12: Adversarial Vulnerability when adding consecutive tasks. The tasks are not weighted.

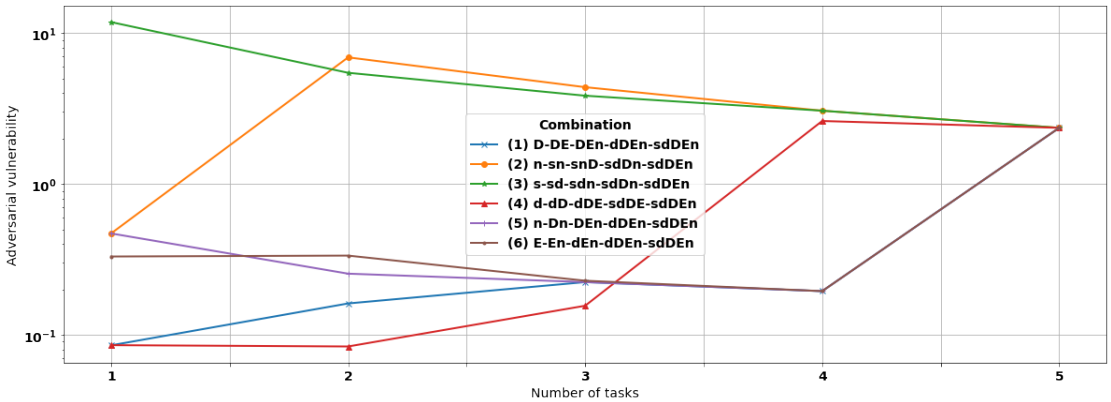


Figure 10.13: Adversarial Vulnerability when adding consecutive tasks. The tasks are weighted (1/N).

Impact of attack settings

Impact of number of steps Figures in Fig 10.15 show the impact of iteration steps on the robustness of the different tasks. The first finding is that while the multi-task models and the mono-task models display similar robustness on one-step and few step attacks, the differences across the models widens as we increase the number of steps (nd attack on a nd model causes a 75% increase in the MSE of z-depth in comparison with a d attack on the same nd model (figure 10.14c).

Especially, some combinations of tasks are more sensitive to the number of iterations. While Ds attack on a Ds model plateau after 15 epochs, D attack on the same Ds model keeps increasing significantly with the number of steps (figure 10.14b). Similarly, Es attack on an Es model plateau after 10 steps, while E attack on the same Es model keeps increasing (figure 10.14d). In general, against the same multi-task model, multi-task attacks tend to plateau much earlier,

Against mono-task attacks, Mono-task models are neither the more robust or the less across the different tasks. For instance training a model with the two tasks d and D makes the model 12% more robust than mono-task D (figure 10.14b, however training the model on the tasks n and D makes the model 48% less robust than the mono-task model D.

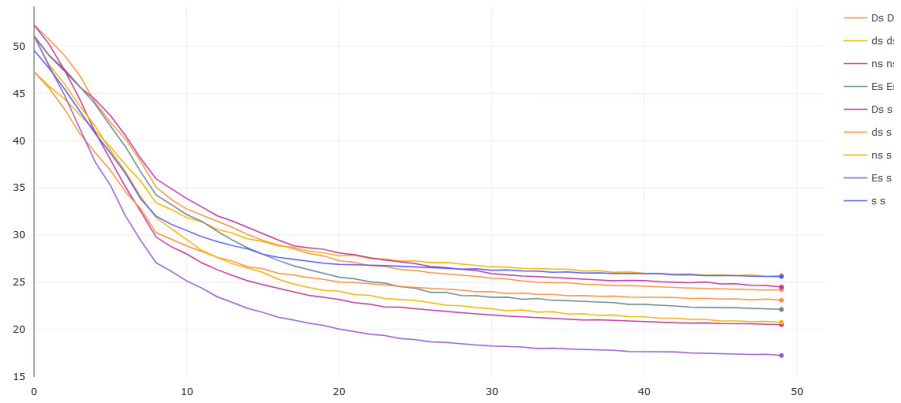
Similar behaviour happens against multi-task attacks. While it is easier to attack a single task model (task D) than attacking 2 tasks together, other combination of tasks are easier to attack than attacking one single task (E and s together are easier to attack than s alone in figure 10.14a). **Conclusion: In general, when given sufficient steps multi-task models perform as poorly as a mono-tasks models and multi-task attacks plateau earlier than their multi-task counterparts.**

Impact of Epsilon We evaluate our different tasks combinations under different strength of attacks. We present the results of 4 tasks in figure 10.16, for each of the main tasks (s,D,E,n), the boxplots reflects the relative error across various auxiliary tasks,both in the mono-task attack context and in the multi-task context. Our results show that strength of attacks impact differently the different tasks. While, the image segmentation task (s) and the Normal prediction (n) task display a linear error with the increased epsilon, the Edge (E) and Depth tasks (D) show an exponential vulnerability to the strength of the attack. This different behaviour reflects both in multi-task attacks and mono-task attacks. It is worth noticing that this different behaviour cannot be explained by the nature of metric used (task S uses mIOu error and cross entropy loss while task n uses MSE and L1 loss) nor the amplitude of error (n and E have closer range of values than E and D).

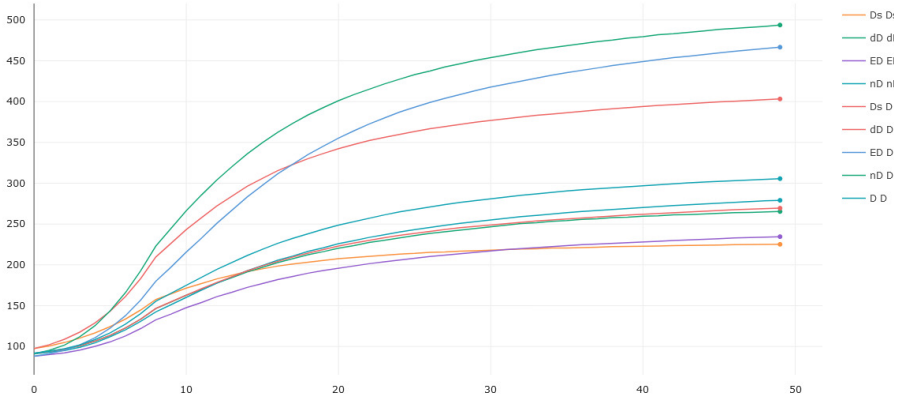
Our results also hint that under highs attack budgets, mono-task attacks and the multi-task attacks achieve close performance, while the variance of robustness provided by the different auxiliary tasks widens.

Impact of Norm We evaluate in this context the impact of using norm L2 in our attacks, under low perturbation amount (ϵ=4), and limited attack steps (25).

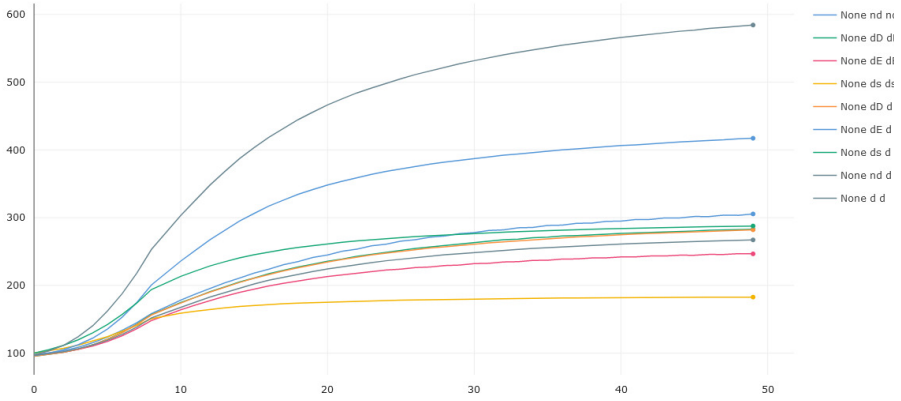
Figure 10.17 shows that the relative task vulerability introduced by an l_2 attack is very limited in comparison with what we can achieve with an l_∞ under the same configuration. **Conclusion: While improving the robustness against l_2 attacks, multi-task learning provides little defense against l_∞ attacks.**



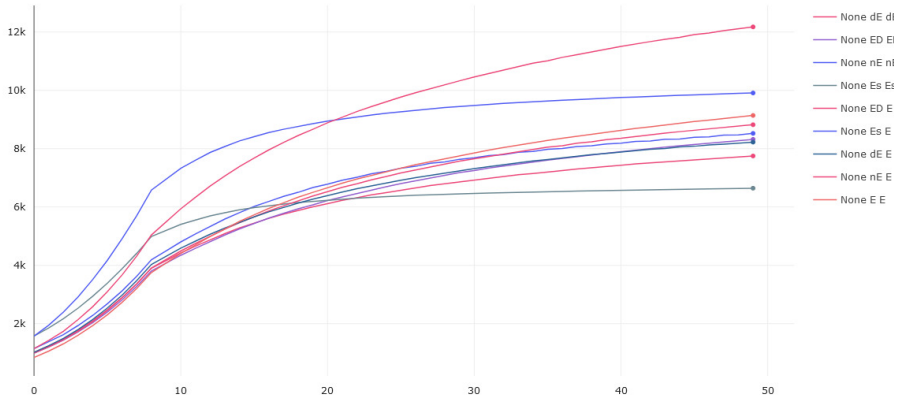
(a) mIoU of Semantic segmentation task



(b) MSE of Euclidian Depth task

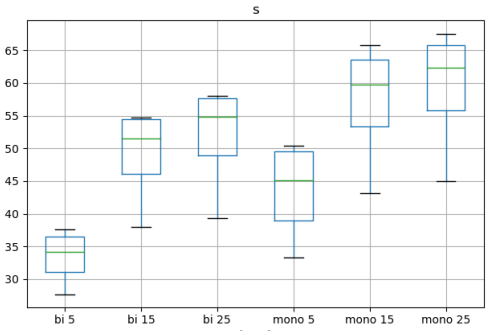


(c) MSE of Z-Depth task

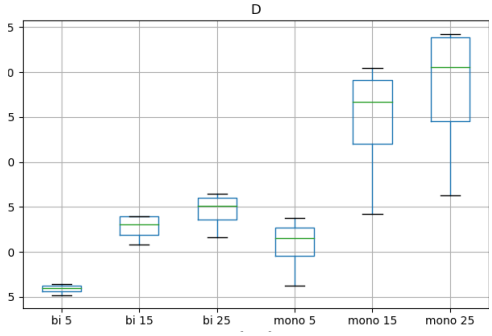


(d) MSE of Edge occlusion task

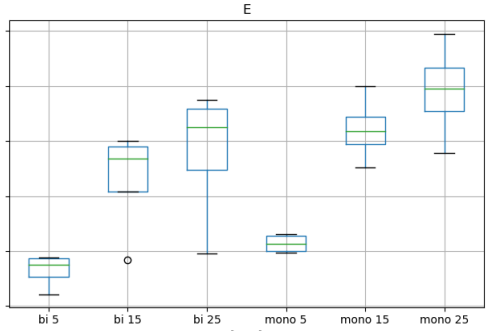
Figure 10.14: Impact of attack steps on the performance of different tasks .Legend: The first letters are the tasks the model has been trained on. The second letters are the tasks that are attacked



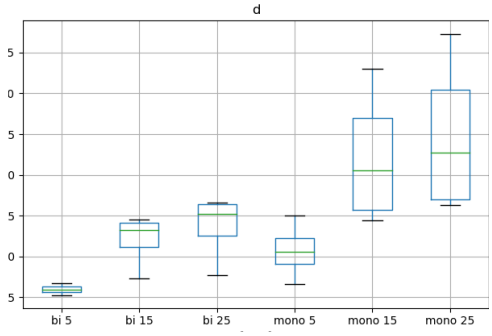
(a) mIoU of Semantic Segmentation



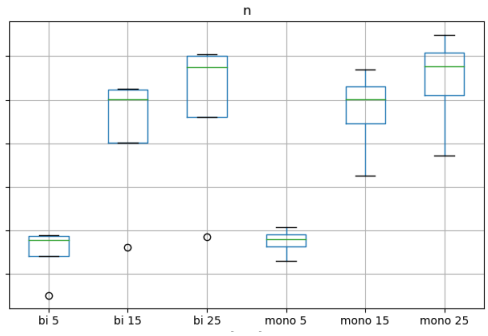
(b) MSE of Euclidian Depth



(c) MSE of Edge Occlusion

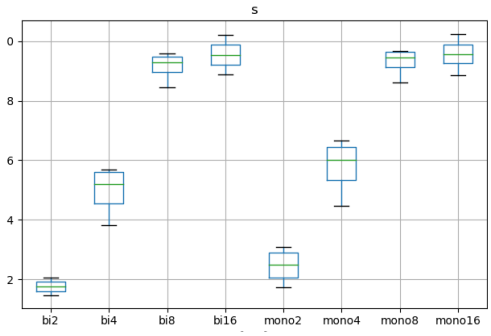


(d) MSE of Z-Depth

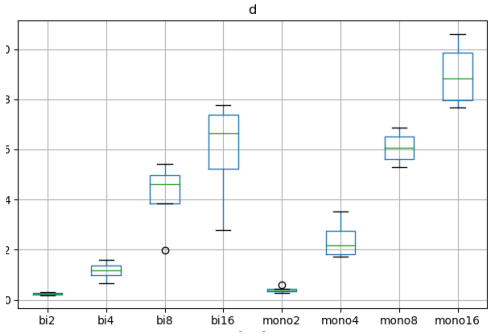


(e) MSE of Normal

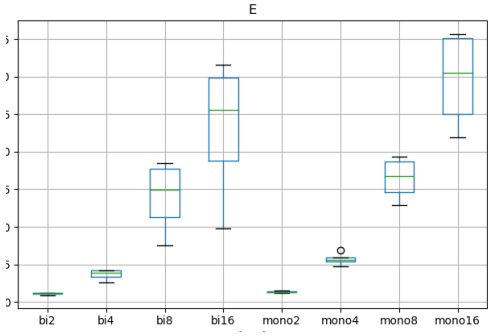
Figure 10.15: Impact of attack steps on the performance of different tasks: We evaluate the relative task robustness of models for 3 different attack steps: 5, 15 and 25; for adversarial attacks against the main task only (mono) or both tasks (multi)



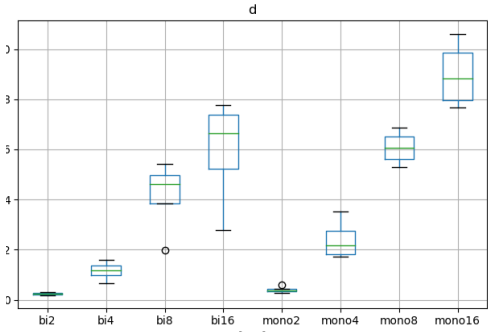
(a) mIoU of Semantic segmentation task



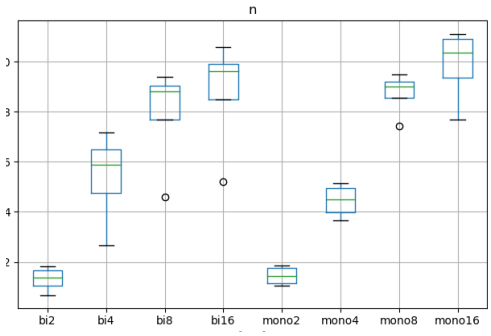
(b) MSE of Euclidian Depth task



(c) MSE of Edge occlusion task

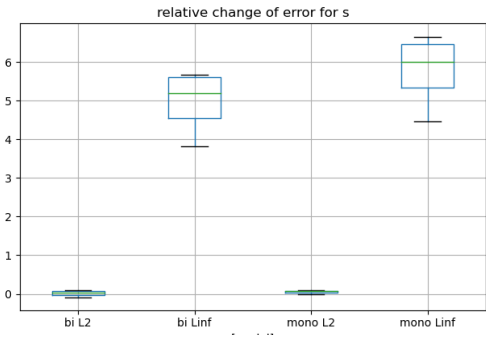


(d) MSE of Z-Depth task

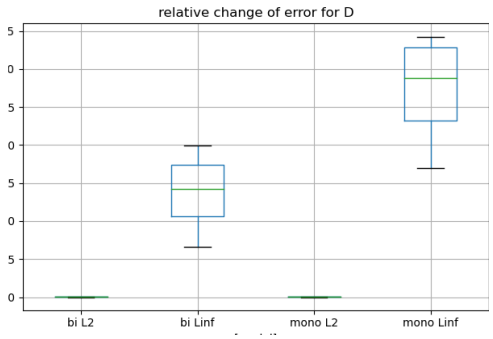


(e) MSE of Normal

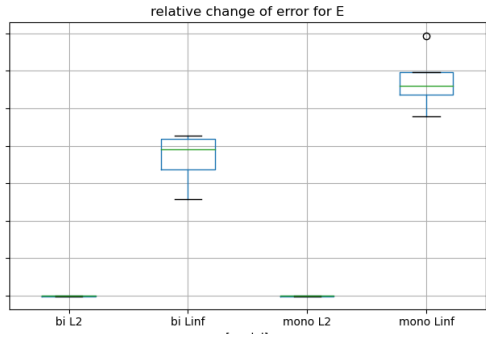
Figure 10.16: Impact of attack strength on the performance of different tasks: We evaluate the relative task robustness of models for different attack budgets ϵ: 2/255; 4/255; 8/255 and 16/255.



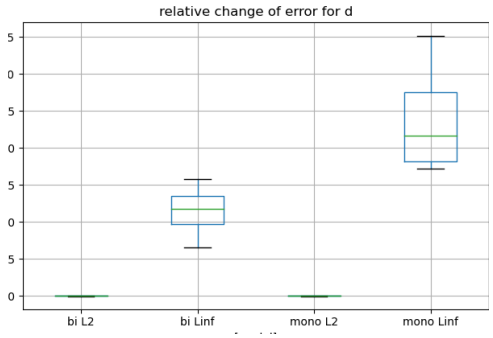
(a) mIoU of Semantic segmentation task



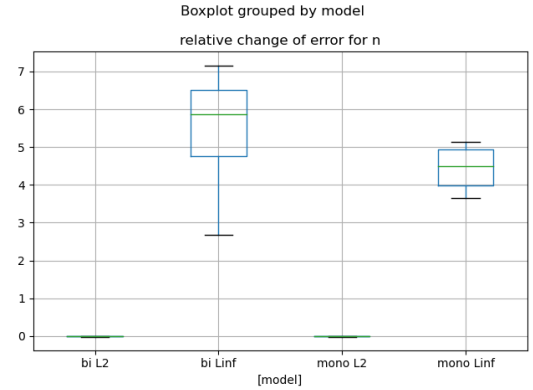
(b) MSE of Euclidian Depth task



(c) MSE of Edge occlusion task



(d) MSE of Z-Depth task



(e) MSE of Normal

Figure 10.17: Impact of attack norm on the performance of different tasks

input : A multi-task model $\mathscr{M}$, an input example x, and $\bar{y} = (y_1, ..., y_M)$ the corresponding ground-truth label for each task $t \in \mathscr{T} = \{t_1, ..., t_M\}$, set of attacked tasks; a step size ϵ_{step}; a maximum perturbation size ϵ; a random sphere size δ; a total iterations L; a number of random restart R; a small value σ to avoid zero divisions.

output : *bestAdv*: The final adversarial example *advX*

$bestRate \leftarrow 0$;
$bestAdv \leftarrow Null$;
for $j \leftarrow 1$ **to** R **do**
  $advX \leftarrow$ `randomSphere(`x, x, δ`)` ;
  $momentum \leftarrow 0$;
  $\mathcal{L}(0) \leftarrow$ `tasksLoss(`$advX$,$\mathscr{M}$, $\mathscr{T}$`)`;
  for $i \leftarrow 1$ **to** L **do**
    $\mathcal{L}(i) \leftarrow$ `tasksLoss(`$advX$,$\mathscr{M}$, $\mathscr{T}$`)` / $(\sigma + \mathcal{L}(0))$;
    $W \leftarrow \mathcal{L}(i) / \mathbb{E}_{t \in \mathscr{T}}[\mathcal{L}_t(i)]$;
    $advX$, $momentum \leftarrow$ `computeAdv(`$advX$, $\bar{y}$, x, ϵ_{step}, ϵ, W, $momentum$`)`
    $success \leftarrow$ `computeSuccess(`$\mathscr{M}$, $advX$, $\bar{y}$`)`;
    if $bestRate \leq success$ **then**
      $bestRate \leftarrow success$;
      $bestAdv \leftarrow advX$;
    end
  end
end

Algorithm 4: Multi-task Weighted Gradient Attack (WGD) algorithm

10.2.9 Appendix D: Algos & Source code

Algorithm

We present in 4 our propsoed WGD algorithm. The adversarial attack computation is detailed in 5 and is very similar to the Momentum attack implementation. The main change of our approach, is the addition of weights to the computed gradients. The weights are computed using the relative inverse attack rate as $r_i(t) = \frac{\tilde{\mathcal{L}}_i(t)}{E_i[\tilde{\mathcal{L}}_i(t)]}$ using the the *Inverse task attack rate* detailed in the main paper.

Source Code

We provide in the review package a folder *Code* with 3 components:
Our source code is under the MIT licence.

input : A set of adversarial images $advX$; $\bar{y}$; a clean example x and its ground truth $\bar{y} = (y_1, ..., y_M)$; W the weight matrix of each task; a maximum perturbation size ϵ; a step perturbation size ϵ_{step}; *momentum* the previous iteration momentum *momentum*; a momentum decay value *decay*

output : $advX$: A more perturbed set of adversarial images; *momentum* the updated momentum computation

$grad \leftarrow W \times$ `LossGradient(`$advX$,$\bar{y}$`)`

$grad \leftarrow \frac{grad}{\|grad\|}$

$grad \leftarrow grad + momentum \times decay$

$momentum \leftarrow grad$

$advX \leftarrow advX + \epsilon_{step} \times grad$

$perturbation \leftarrow$ `Project(`$advX$ - x,ϵ`)`

$advX \leftarrow x + perturbation$

Algorithm 5: *computeAdv* procedure

- **MTRobust**: cloned from `https://github.com/columbia/MTRobust/` and extended with additional models (Xception, WideResnet). This repository is used to train the models following the same setting as the original paper of MTRobust. Read their documentation for more instructions about how to train models. Or use our scripts in *MTRobust/jobs/*.
  You will need to download the Taskonomy dataset as explained in the original repository and update the configuration files in *MTRobust/jobs/*.
- **MTVulnerability**: Our package to attack and evaluate the vulnerability of the models. The folder *MTVulnerability/jobs* contains the script you can run directly. Please read the specific *README* file of our package for more details & instructions.
- **models**: This folder provides one pretrained Taskonomy model for task combination s (semantic segmentation) and d (Z-depth) to use as a quick test. You can use this folder for the variable **MODEL** in the scripts located in *MTVulnerability/jobs*.

Our experiments use CometML to track and record the results of our experiments. You will need a valid (free) account from `https://www.comet.ml/` and a personal API Key to send the results of the experiments.

10.3 ATTA: Improving Adversarial Training with Task Augmentation.

10.3.1 Appendix A: Replication

Datasets

X-ray datasets We show in table 10.7 the general properties of the datasets used in training our models. Table 10.16 ([CHB+20]) details the number of positive and negative examples with each label for each dataset. Our models are trained either on CheXpert or NIH depending on the evaluation.

Our evaluation covers as target tasks very scarce pathologies (Edema, Pneumonia), and medium scarce pathologies (Atelectasis), across both datasets.

	NIH	CheXpert
Number of patient radiographs	112,120	224,316
Number of patients	30,805	65,240
Age in years: mean (standard deviation)	46.9 (16.6)	60.7 (18.4)
Percentage of females (%)	43.5%	40.6%
Number of pathology labels	8	14

Table 10.7: Characteristics of NIH and CheXpert datasets used in our evaluation.

Architectures

The majority of the tests are carried out using the Resnet50v2 [HZR+16] encoder, which has a depth of 50 and 25.6M parameters. This encoder is the main focus because it is the most widely used for Xray image classification [GRL+19]. We also perform some tests using the WRN-28-10[ZK16] encoder, which has a depth of 28, a width multiplier of 10, and 36M parameters.

Adversarial Training

The outer minimization: We use MADRY adversarial training [MMS+17c], i.e. we train the model using a summed loss computed from the clean and adversarial examples. for ATTA, we use a backpropagation over the pareto optimal of the four losses. The learning uses the SGD optimizer with lr=0.1, a cosine annealing, and checkpoint over the best performance.

The inner maximization: We generate the adversarial examples with PGD [MMS+17b], on ℓ_∞ norms and $\epsilon = 8/255$ for CIFAR-10 and STL-10 and $\epsilon = 4/255$ for CheXpert and NIH models. We use in the iterative attack 1 random start, and 10 steps.

Dataset	NIH	CheXpert
Atelectasis	1702/29103	12691/14317
Cardiomegaly	767/30038	9099/17765
Consolidation	427/30378	5390/22504
Edema	82/30723	14929/20615
Effusion	1280/29525	20640/23500
Emphysema	265/30540	-
Enlarged Cardio	-	5181/20506
Fibrosis	571/30234	-
Fracture	-	4250/14948
Hernia	83/30722	-
Infiltration	3604/27201	-
Lung Lesion	-	4217/14422
Lung Opacity	-	30873/15675
Mass	1280/29525	-
Nodule	1661/29144	-
Pleural Thickening	763/30042	-
Pneumonia	168/30637	2822/14793
Pneumothorax	269/30536	4311/32685

Table 10.8: Samples distributions across each pathology and dataset. Each cell shows the number of positive/negative samples of the label. There are 7 common pathologies in NIH and CheXpert datasets. Among those, in bold the pathologies evaluated as target task, and in underline the pathologies used as an auxiliary.

Robustness evaluation

We evaluate the robustness against PGD-10 on ℓ_∞ norms and $\epsilon = 8/255$ for CIFAR-10 and STL-10 and $\epsilon = 4/255$ for CheXpert and NIH models. We also evaluate CIFAR-10 models against AutoAttack [CH20a]. Autoattack is a mixture of $\ell_\infty epsilon = 8/255$ attacks: untargeted AUTOPGD (a variant of PGD with an adaptive step) on the cross-entropy loss with 100 steps, targeted AUTOPGD with 100 steps, a 100 steps FAB attack, and finally a 5000 queries Square attack.

These hyper-parameters of AutoAttack are consistent with AutoAttack's default parameterization in kim2020torchattacks, croce2020robustbench.

Computation budget

We train all our models on slurm nodes, using single node training. Each node has one A100 GPU 32Gb V100 SXM2. We train CIFAR-10 and STL-10 models for 400 epochs and CheXpert and NIH models for 200 epochs. The WRN-70-16 model is trained for 40 epochs to account for being 10 times larger than the Resnet50 used for the main evaluation.

Source code

The source code is provided with the zip on OpenReview submissions. Pretrained models are available at `https://figshare.com/projects/ATTA/139864` Refer to **README.md** for installation instructions.

Our license is **MIT Licence**, and we use the following external packages:

Torchxrayvision: Located in folder **./torchxrayvision**. Adapted from `https://github.com/mlmed/torchxrayvision`: Apache Licence

Taskonomy/Taskgrouping: Located in folder **./utils/multitask_models**. Adapted from `https://github.com/tstandley/taskgrouping/` MIT Licence

LibMTL: Located in folder **./utils/weights**. Adapted from `https://github.com/median-research-group/LibMTL` MIT Licence

10.3.2 Appendix B: Detailed results of the main study

Limited data training with CIFAR-10

ATTA: To evaluate whether ATTest accuracy (i.e. equal weights task augmentation) is effective when access to adversarial training data is limited, we first train models with the full dataset for 200 epochs then we adversarial fine-tune (PGD-4; 8/255) the models with a subset of training data (10%, 50%). For each scenario, we fine-tune 3 different models with different seeds and report in Table 10.9 the Test Accuracy (Test accuracy) and Robust Accuracy (Robust accuracy) with and without an auxiliary task. We report the mean and standard deviation across the runs. The std across the experiments is pretty low and the conclusions of the main paper hold.

W-ATTA: With the MGDA weighting strategy, we train the models for 400 epochs using one 10%, 25%, and 50% of the data and 3 auxiliary strategies, and evaluate the clean and robust performance in Table 10.10. W-ATTA outperforms single task adversarial training for the cases with 10% and 25% of the total training data.

CheXpert detailed results

We extend the evaluation of the main paper to 6 additional combinations of auxiliary tasks and target task, using the **Pneumonia** pathology as a a target. We present all the results in Table 10.11. These extended results corroborate that adversarial training with auxiliary task significiantly improves the robustness of classification models on the CheXpert dataset [IRa19].

Table 10.9: Evaluation results of 4 Different $(\mathcal{D}_i, \mathcal{T}_i, \mathcal{A}_i)$ Scenarios: $\mathcal{D}_1$ (adversarial fine-tuning with 10% of the training data), $\mathcal{D}_2$ (adversarial fine-tuning with 50% of the training data), $\mathcal{T}_{1,2,3}$ training respectively without an auxiliary task, with Rotation and with Jigsaw task, $\mathcal{A}_1$ (Robust Accuracy against a PGD-4 attack), $\mathcal{A}_2$ (Robust Accuracy against a PGD-10 attack).

Dataset subset	Auxiliary	PGD steps	Metric	mean	std
0.1	*None*	10	Test accuracy	60.41	0.62
			Robust accuracy	8.37	0.32
		4	Test accuracy	60.38	0.59
			Robust accuracy	11.81	0.23
	Jigsaw	10	Test accuracy	51.98	0.53
			Robust accuracy	32.41	0.46
		4	Test accuracy	51.06	1.47
			Robust accuracy	32.26	0.85
	Rotation	10	Test accuracy	50.41	0.11
			Robust accuracy	15.01	0.29
		4	Test accuracy	50.17	0.49
			Robust accuracy	20.01	0.27
	Macro	10	Test accuracy	65.62	0.48
			Robust accuracy	22.42	0.35
		4	Test accuracy	65.65	0.42
			Robust accuracy	42.68	0.38
0.5	*None*	10	Test accuracy	77.45	0.25
			Robust accuracy	25.04	0.15
		4	Test accuracy	77.51	0.15
			Robust accuracy	31.71	0.08
	Jigsaw	10	Test accuracy	59.72	0.45
			Robust accuracy	29.08	1.58
		4	Test accuracy	58.42	1.16
			Robust accuracy	33.68	0.52
	Rotation	10	Test accuracy	59.77	0.58
			Robust accuracy	17.09	0.51
		4	Test accuracy	59.69	0.70
			Robust accuracy	24.56	0.49
	Macro	10	Test accuracy	73.68	0.72
			Robust accuracy	33.76	0.71
		4	Test accuracy	73.62	0.81
			Robust accuracy	54.14	0.63

Table 10.10: W-ATTA different data scenarios: 10%, 25% and 50% of CIFAR-10 dataset. We evaluate 3 different task augmentations with MGDA weighting strategy.

Scenario	Clean accuracy (%)				Robust accuracy (%)			
	None	*Jigsaw*	*Macro*	*Rotation*	*None*	*Jigsaw*	*Macro*	*Rotation*
10%	52.66	42.7	54.89	47.07	12.46	32.14	13.43	39.2
25%	68.39	49.76	68.54	62.85	24.56	32.08	27.74	47.75
50%	76.13	66.57	76.5	78.19	33.69	23.79	31.94	16.63

Table 10.11: Robust and clean AUC of CheXpert models trained with ATTA.

Target Task	Auxiliary Task	Robust AUC	Clean AUC
Atelectasis	Single task	50.00	58.76
Atelectasis	Cardiomegaly	71.20	71.97
Atelectasis	Pneumothorax	70.93	71.92
Atelectasis	Age	46.26	66.89
Atelectasis	Gender	83.00	83.35
Atelectasis	Jigsaw	63.81	65.92
Atelectasis	Rotation	78.32	74.50
Edema	Single task	55.69	52.42
Edema	Cardiomegaly	52.74	55.79
Edema	Pneumothorax	47.40	58.86
Edema	Age	59.17	53.41
Edema	Gender	31.46	56.07
Edema	Jigsaw	70.47	67.77
Edema	Rotation	52.59	55.98
Pneumonia	Single task	38.70	56.66
Pneumonia	Cardiomegaly	57.47	57.05
Pneumonia	Pneumothorax	32.25	56.74
Pneumonia	Age	49.15	56.58
Pneumonia	Gender	60.08	57.59
Pneumonia	Jigsaw	46.45	56.47
Pneumonia	Rotation	60.76	60.00

Table 10.12: Four Different $\mathcal{T}_i$ Scenarios: $\mathcal{T}_1$; standard training, $\mathcal{T}_2$: adversarial training @ Goodfellow, $\mathcal{T}_3$: adversarial training @ Madry, $\mathcal{T}_4$: adversarial training @ Trades [ZYJ+19], and $\mathcal{T}_5$: adversarial training @ Fast[WRK20], with 3 different task augmentations and equal weighting strategies.

Scenario	Clean accuracy (%)			Robust accuracy (%)		
	Jigsaw	*Macro*	*Rotation*	*Jigsaw*	*Macro*	*Rotation*
$\mathcal{T}_1$: Standard training	88.78	93.04	69.67	0.59	0.06	3.18
$\mathcal{T}_3$: Madry AT	64.9	**83.00**	68.23	20.25	32.16	25.43
$\mathcal{T}_2$: Goodfellow AT	55.01	77.49	42.29	40.5	38.24	34.43
$\mathcal{T}_4$: Trades AT	46.4	60.73	50.24	33.61	**42.76**	42.05
$\mathcal{T}_5$: Fast AT	52.35	78.18	56.36	19.06	26.56	19.84

10.3.3 Appendix C: Complementary results

ATTA on a supplementary Chest X-ray dataset: NIH

We present in Figure 10.18 similar study of the main paper, but on the NIH dataset. Our conclusions that ATTA outperforms Adversarial training (circles in 10.18) are confirmed on this dataset as well.

ATTA combined with other adversarial training strategies

While the default setup of ATTA follows Madry adversarial training, i.e., we backpropagate both the loss from the clean and adversarial examples, we compare in Table 10.12 different adversarial training strategies with ATTA. These results motivate our choice of Madry stratrgy: All other adversarial training strategies degrade significantly the clean accuracy to unacceptable levels.

ATTA combined with other weighting strategies

We evaluate 5 weighting strategies on Resnet-50 architectures:

1. Equal weights (Equal),
2. Impartial Multi-task Learning (IMTL) [LLK+21],
3. Multiple Gradient Descent Algorithm (MGDA) [SK18],
4. Gradient Vaccine (GradVac) [WTF+20],
5. Project Conflicting Gradients (PCGrad) [YKG+20]

The results in Table 10.13 uncover that adversarial training using the Macro task yields the best performance in 4 over 5 weighting strategies, and that MGDA weighting strategies yields the best clean and robust accuracy among the weighting

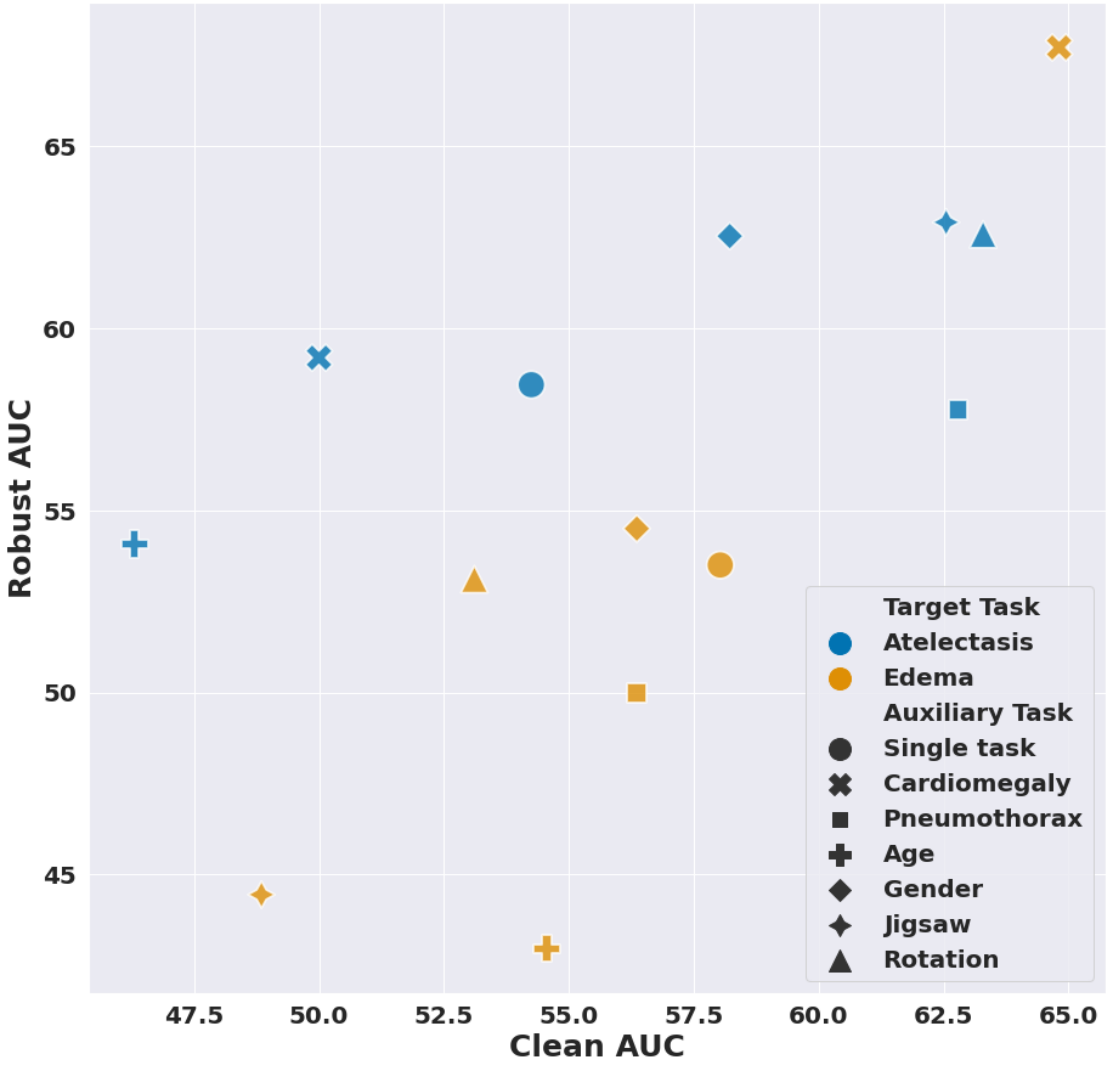


Figure 10.18: Comparison of different Task Augmentation strategies with single-task models using Adversarial Training; Clean and robust AUC of ATTA vs Single task adversarial training to diagnose Atelectasis and Edema pathologies for the NIH dataset

Table 10.13: Evaluation results of Two Different $\mathcal{T}_i$ Scenarios: $\mathcal{T}_1$ (standard training), $\mathcal{T}_2$ (adversarial training), with 3 different task augmentations and 5 weighting strategies. In bold, the best values for each scenario

Scenario	Weight	Clean accuracy (%)			Robust accuracy (%)		
		Jigsaw	*Macro*	*Rotation*	*Jigsaw*	*Macro*	*Rotation*
$\mathcal{T}_1$	Equal	88.78	93.04	69.67	0.59	0.06	3.18
	GradVac	89.08	93.01	68.42	0.26	0.09	3.81
	IMTL	61.46	93.75	71.24	0.98	0.09	3.81
	MGDA	41.65	**93.89**	70.26	0.00	0.24	**4.33**
	PCGrad	88.85	92.99	69.11	0.69	0.11	3.13
$\mathcal{T}_2$	Equal	55.01	**77.49**	42.29	40.5	38.24	34.43
	GradVac	44.67	64.11	57.71	36.24	35.56	40.17
	IMTL	42.05	69.63	59.61	33.84	48.21	39.93
	MGDA	43.99	73.7	56.51	32.95	**48.38**	36.13
	PCGrad	41.6	63.76	56.38	33.8	44.74	41.59

strategies (with the **MACRO** task). MGDA uses multi-objective optimization to converge to the pareto-stationnary for both the tasks we train over. This search algorithm shows that we can attain loss landscapes with high clean and robust performances that greedy gradient algorithms (equal weights, GradVac, PCGrad) fail to uncover.

We used the default hyper-parameters for the weighting strategies. One possible work would be to fine-tune the weighting strategies to the adversarial training setting.

Adaptive attacks: AutoAttack

We evaluate for all the models of the study the adversarial robustness against AutoAttack. For 3/4 scenarios, adversarial training with task augmentation using **Macro** tasks outperforms single-task adversarial training.

Surrogate attacks

We evaluate in Table 10.15 the transferability of attacks from a surrogate model to a target model. Both models are trained on the same training dataset.

(1) When the target model has an auxiliary task, the success rate of the attack crafted from a single-task surrogate model drops by 14%. (2) When the surrogate model has an auxiliary task, the success rate against a single task target model drops by 60%.

(1) indicates that the adversarial examples generated to fool a multi-task model actually lie in a loss landscape that is not adversarial for the single task model:

Table 10.14: Robust accuracy (%) against AutoAttack of different models adversarially trained with ATTA, with 3 different task augmentations, compared to their counterpart single task adversarially trained models. In bold the cases where ATTA outperforms single-task AT.

Dataset	Scenario	Auxiliary task			
		None	*Jigsaw*	*Macro*	*Rotation*
CIFAR-10	100% Dataset	27.01	**29.63**	**32.54**	13.82
	10% Dataset	14.27	11.63	**15.00**	11.56
	WideResnet28-10	36.29	15.99	34.44	25.72
STL-10	100% Dataset	19.40	12.78	**20.02**	17.64

Table 10.15: Evaluation results of Three Different combinations of surrogate models and target models. For each combination, we craft the adversarial examples on the surrogate and evaluate the success rate of the examples on the target models. Both surrogate and target models are trained with standard training.

Target →	Single Task	Auxiliary *Rotation*	Auxiliary *Jigsaw*
Surrogate ↓	Success rate %		
Single Task	98.47	84.55	86.56
Auxiliary *Rotation*	37.95	98.05	79.86
Auxiliary *Jigsaw*	37.48	79.35	98.59

The PGD optimization is misguided when multiple tasks are present.

(2) The adversarial examples generated against one single task are actual relevant to models with multiple tasks. It means that multitask learning by itself has the same vulnerable area as the single task-learning, it is just that gradient-based attacks have more difficulty to find them.

10.4 Towards Generalizable Machine Learning for Chest X-ray Diagnosis with Multi-task learning.

10.4.1 Appendix A: Experimental protocol details

Datasets We show in table 10.7 the general properties of the datasets used in training our models. Table 10.16 ([CHB$^+$20]) details the number of positive and negative examples with each label for each dataset. Our models are trained either on NIH, PC or CheXpert depending of the evaluation. Models *C1* and *C2* are trained on CheXpert dataset and model *A* is trained on all the 8 datasets.

Dataset	NIH	PadChest	CheXpert	Google	MIMIC_CH	MIMIC_NB	OpenI	Kaggle
Atelectasis	1702/29103	2441/59674	12691/14317	-	4077/30954	4048/32058	271/2996	-
Cardiomegaly	767/30038	5390/56725	9099/17765	-	3743/32312	3275/33431	185/3082	-
Consolidation	427/30378	494/61621	5390/22504	-	816/32297	762/33564	-	-
Edema	82/30723	108/62007	14929/20615	-	1157/33610	1121/34731	50/3217	-
Effusion	1280/29525	1637/60478	20640/23500	-	3713/33401	3595/34489	120/3147	-
Emphysema	265/30540	546/61569	-	-	-	-	84/3183	-
Enlarged Cardio	-	-	5181/20506	-	692/31505	660/32641	-	-
Fibrosis	571/30234	341/61774	-	-	-	-	17/3250	-
Fracture	-	1665/60450	4250/14948	60/1635	972/30961	696/32320	78/3189	-
Hernia	83/30722	988/61127	-	-	-	-	41/3226	-
Infiltration	3604/27201	4438/57677	-	-	-	-	66/3201	-
Lung Lesion	-	-	4217/14422	-	1321/31033	1271/32187	3/3264	-
Lung Opacity	-	-	30873/15675	601/1094	5426/31175	5301/32371	327/2940	9555/20672
Mass	1280/29525	507/61608	-	-	-	-	6/3261	-
Nodule	1661/29144	2194/59921	-	-	-	-	68/3199	-
Pleural Thickening	763/30042	2076/60039	-	-	-	-	30/3237	-
Pneumonia	168/30637	2051/60064	2822/14793	-	2176/33347	2042/34479	68/3199	9555/20672
Pneumothorax	269/30536	98/62017	4311/32685	72/1623	560/33651	500/34760	14/3253	-

Table 10.16: Samples distributions across each pathology and dataset. Each cell shows the number of positive/negative samples of the label. In bold the pathologies we use in training our models. Those are the 7 common pathologies in NIH, PC and Chexpert datasets.

Models We provide in table 10.18 the test and generalization performances of models trained on the *CheXpert* dataset obtained with Auxiliary (Aux) pathology learning compared to fine-tuning (Tune), single pathology learning (Single) and all pathology learning (All). The table presents absolute AUC values.

Table 10.19, table 10.20 and table 10.21 show the test AUC of models trained and evaluated on the same dataset. Respectively, on the dataset *CheXpert*, *NIH*, and *PadChest*.

We provide in table 10.22 and table 10.23 the generalization performances of models trained on the *CheXpert* dataset obtained with Auxiliary (Aux) pathology learning and tested on respectively NIH and PadChest datasets.

Pathology 1 (p1)	Pathology 2 (p2)	Both positive	Both negative	p1=1 p2=0	p1=0 p2=1
Atelectasis	Edema	179	5526	442	9
Atelectasis	Effusion	1505	5569	537	30
Atelectasis	Cardiomegaly	342	5520	212	15
Atelectasis	Consolidation	177	5589	513	7
Atelectasis	Pneumothorax	268	5530	720	5
Atelectasis	Pneumonia	105	5512	50	5
Edema	Atelectasis	179	5526	9	442
Edema	Effusion	825	7266	153	642
Edema	Cardiomegaly	558	5882	41	806
Edema	Consolidation	70	6705	122	97
Edema	Pneumothorax	41	6464	119	49
Edema	Pneumonia	77	5588	22	83
Effusion	Atelectasis	1505	5569	30	537
Effusion	Edema	825	7266	642	153
Effusion	Cardiomegaly	1029	6513	299	840
Effusion	Consolidation	638	8386	409	139
Effusion	Pneumothorax	687	8128	1194	87
Effusion	Pneumonia	282	5580	52	161
Cardiomegaly	Atelectasis	342	5520	15	212
Cardiomegaly	Edema	558	5882	806	41
Cardiomegaly	Effusion	1029	6513	840	299
Cardiomegaly	Consolidation	143	6268	665	76
Cardiomegaly	Pneumothorax	67	6211	546	55
Cardiomegaly	Pneumonia	87	5534	58	49
Consolidation	Atelectasis	177	5589	7	513
Consolidation	Edema	70	6705	97	122
Consolidation	Effusion	638	8386	139	409
Consolidation	Cardiomegaly	143	6268	76	665
Consolidation	Pneumothorax	54	7113	179	51
Consolidation	Pneumonia	192	5540	8	175
Pneumothorax	Atelectasis	268	5530	5	720
Pneumothorax	Edema	41	6464	49	119
Pneumothorax	Effusion	687	8128	87	1194
Pneumothorax	Cardiomegaly	67	6211	55	546
Pneumothorax	Consolidation	54	7113	51	179
Pneumothorax	Pneumonia	20	5558	6	103
Pneumonia	Atelectasis	105	5512	5	50
Pneumonia	Edema	77	5588	83	22
Pneumonia	Effusion	282	5580	161	52
Pneumonia	Cardiomegaly	87	5534	49	58
Pneumonia	Consolidation	192	5540	175	8
Pneumonia	Pneumothorax	20	5558	103	6

Table 10.17: Pathology co-occurence for Chexpert datasets.

	Test AUC % on CheXpert				Test AUC % on NIH			
Path	Fine-tuning	Auxiliary Task Learning	Single	All	Fine-tuning	Auxiliary Task Learning	Single	All
Atelectasis	86.88	89.59	87.97	89.17	67.10	69.25	70.77	67.82
Cardiomegaly	85.12	89.71	89.16	89.09	67.84	76.88	75.40	70.67
Consolidation	84.97	86.74	85.35	88.61	68.56	71.13	66.52	66.49
Edema	88.55	91.57	90.64	90.64	72.85	75.91	69.88	68.54
Effusion	90.84	93.52	93.73	91.96	81.02	83.15	83.66	79.81
Pneumonia	77.29	80.06	77.73	83.42	60.81	62.50	59.85	67.78
Pneumothorax	80.25	83.42	81.47	83.71	67.34	68.17	58.69	60.08

Table 10.18: AUC Performance of models trained on each pathology on the CheXpert dataset, tested on CheXpert (middle) and tested on NIH (right). Tune represents the fine-tuned model; Aux when learned with an Auxiliary ; Single represents a model training on a single pathology; and All when all the pathologies are learnt at once. Aux and Tune report only the best performing models.

Main pathology Auxiliary	ATE	CAR	CON	EDE	EFF	PNE	PNX	AVG
Atelectasis	0.88	0.90	0.87	0.91	0.93	0.80	0.82	0.87
Cardiomegaly	0.89	0.89	0.87	0.92	0.93	0.78	0.81	0.87
Consolidation	0.89	0.90	0.85	0.89	0.93	0.79	0.83	0.87
Edema	0.89	0.89	0.84	0.91	0.93	0.79	0.81	0.87
Effusion	0.90	0.89	0.86	0.91	0.94	0.80	0.83	0.87
Pneumonia	0.88	0.89	0.86	0.91	0.92	0.78	0.80	0.86
Pneumothorax	0.89	0.89	0.86	0.91	0.94	0.80	0.81	0.87
Average	0.89	0.89	0.86	0.91	0.93	0.79	0.82	0.87

Table 10.19: Test AUC of models trained on CheXpert and evaluated on CheXpert

We provide in table 10.24 and table 10.25 the generalization performances of models trained on the *NIH* dataset obtained with Auxiliary (Aux) pathology learning and tested on respectively CheXpert and PadChest datasets.

Main pathology / Auxiliary	ATE	CAR	CON	EDE	EFF	PNE	PNX	AVG
Atelectasis	0.79	0.92	0.75	0.85	0.88	0.65	0.85	0.81
Cardiomegaly	0.77	0.91	0.74	0.76	0.87	0.65	0.85	0.79
Consolidation	0.77	0.92	0.72	0.81	0.87	0.66	0.84	0.80
Edema	0.77	0.90	0.72	0.69	0.87	0.58	0.84	0.77
Effusion	0.80	0.91	0.75	0.79	0.87	0.66	0.85	0.81
Pneumonia	0.78	0.91	0.71	0.72	0.87	0.59	0.83	0.77
Pneumothorax	0.77	0.91	0.74	0.78	0.87	0.64	0.84	0.79
Average	0.78	0.91	0.73	0.77	0.87	0.63	0.84	0.79

Table 10.20: Test AUC of models trained on NIH and evaluated on NIH

Main pathology / Auxiliary	ATE	CAR	CON	EDE	EFF	PNE	PNX	Average
Atelectasis	0.80	0.94	0.83	0.96	0.95	0.79	0.82	0.88
Cardiomegaly	0.82	0.94	0.86	0.97	0.94	0.81	0.83	0.88
Consolidation	0.76	0.94	0.81	0.96	0.93	0.78	0.78	0.85
Edema	0.71	0.94	0.82	0.96	0.94	0.76	0.78	0.85
Effusion	0.82	0.94	0.85	0.97	0.95	0.79	0.82	0.87
Pneumonia	0.80	0.94	0.84	0.96	0.95	0.79	0.84	0.87
Pneumothorax	0.72	0.93	0.82	0.96	0.94	0.75	0.78	0.83
Average	0.78	0.94	0.83	0.96	0.94	0.78	0.81	0.86

Table 10.21: Test AUC of models trained on PC and evaluated on PC

Main pathology / Auxiliary	ATE	CAR	CON	EDE	EFF	PNE	PNX
Atelectasis	0.66	0.76	0.67	0.71	0.81	0.60	0.65
Cardiomegaly	0.69	0.75	0.68	0.63	0.80	0.58	0.65
Consolidation	0.70	0.76	0.66	0.73	0.82	0.62	0.69
Edema	0.69	0.72	0.70	0.69	0.81	0.58	0.61
Effusion	0.69	0.72	0.72	0.58	0.82	0.59	0.63
Pneumonia	0.69	0.72	0.69	0.65	0.79	0.61	0.69
Pneumothorax	0.68	0.72	0.68	0.67	0.78	0.58	0.60

Table 10.22: Test AUC of models trained on CheXpert and evaluated on NIH

Main pathology Auxiliary	ATE	CAR	CON	EDE	EFF	PNE	PNX
Atelectasis	0.69	0.87	0.74	0.91	0.90	0.60	0.63
Cardiomegaly	0.69	0.84	0.78	0.92	0.90	0.63	0.59
Consolidation	0.67	0.87	0.72	0.94	0.91	0.65	0.69
Edema	0.68	0.84	0.77	0.93	0.91	0.61	0.62
Effusion	0.70	0.86	0.81	0.92	0.91	0.55	0.66
Pneumonia	0.69	0.84	0.74	0.94	0.85	0.55	0.63
Pneumothorax	0.66	0.84	0.76	0.94	0.90	0.60	0.78

Table 10.23: Test AUC of models trained on CheXpert and evaluated on PC

Main pathology Auxiliary	ATE	CAR	CON	EDE	EFF	PNE	PNXx
Atelectasis	0.72	0.72	0.77	0.56	0.84	0.55	0.48
Cardiomegaly	0.65	0.64	0.78	0.66	0.60	0.58	0.52
Consolidation	0.77	0.82	0.66	0.67	0.79	0.60	0.60
Edema	0.59	0.76	0.65	0.68	0.69	0.58	0.51
Effusion	0.80	0.59	0.74	0.63	0.81	0.61	0.59
Pneumonia	0.77	0.75	0.70	0.76	0.74	0.58	0.67
Pneumothorax	0.50	0.64	0.67	0.49	0.76	0.52	0.60

Table 10.24: Test AUC of models trained on NIH and evaluated on CheXpert

Main pathology Auxiliary	ATE	CAR	CON	EDE	EFF	PNE	PNXx
Atelectasis	0.75	0.89	0.76	0.74	0.91	0.66	0.74
Cardiomegaly	0.73	0.87	0.81	0.89	0.87	0.62	0.80
Consolidation	0.74	0.90	0.71	0.79	0.92	0.63	0.73
Edema	0.73	0.89	0.68	0.79	0.89	0.55	0.77
Effusion	0.77	0.87	0.77	0.84	0.89	0.64	0.75
Pneumonia	0.73	0.90	0.70	0.85	0.87	0.54	0.75
Pneumothorax	0.73	0.88	0.71	0.57	0.90	0.55	0.70

Table 10.25: Test AUC of models trained on NIH and evaluated on PC

	CHEX → CHEX						CHEX → NIH					
	mean	std	min	max	single	all	mean	std	min	max	single	all
ATE	88.83	0.57	87.80	89.59	87.97	89.17	68.39	0.52	67.73	69.25	70.77	67.82
CAR	89.21	0.37	88.76	89.71	89.16	89.09	74.08	1.87	71.90	76.88	75.40	70.67
CON	86.12	0.78	84.50	86.74	85.35	88.61	67.76	2.00	65.04	71.13	66.52	66.49
EDE	90.78	0.73	89.43	91.57	90.64	90.60	64.17	6.06	57.05	75.91	69.88	68.54
EFF	92.93	0.47	92.08	93.52	93.73	91.96	81.66	1.26	80.15	83.15	83.66	79.81
PNE	79.39	0.72	78.10	80.06	77.73	83.42	58.90	1.93	56.99	62.50	59.85	67.78
PNX	81.85	1.19	80.16	83.42	81.47	83.71	63.91	3.29	59.71	68.17	58.69	60.08

Table 10.26: Statistic of AUC performance computed for different combinations of models trained on the CheXpert dataset and evaluated on CheXpert (left) and NIH (right)

10.4.2 Appendix B: Pathology selection has a significant impact on generalization

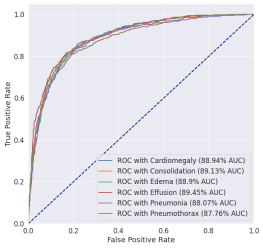
Evaluation of Resnet50 architectures (details of the main paper) We present in ROC curves for all pathologies for models:

- trained on CheXpert and evaluated on NIH in figure 10.19;
- trained on CheXpert and evaluated on PC in figure 10.20;
- trained on NIH and evaluated on CheXpert in figure 10.21;
- trained on NIH and evaluated on PC in figure 10.22;

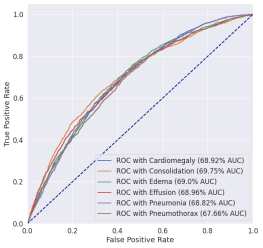
For each figure, we provide for reference the test performance on the original dataset using in the training, then the test performance on the target dataset.

We compute the statistics of mean, standard deviation, maximum and minimum across all these values and present it in table 10.26.

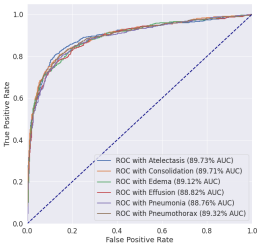
Evaluation of other architectures We run the same experiments as Section 7.3, but using a Densenet121 architecture. Figure 10.23 shows that our claims, evaluated on Resnet50 in the main paper are confirmed on other architectures. While the AUC of Edema shows little variance across combinations of auxiliary pathologies on the source dataset (CheXpert, Figure 10.23i), the choice of the auxiliary pathology can have a significant impact on the AUC performance on the target dataset (NIH , Figure 10.23j).



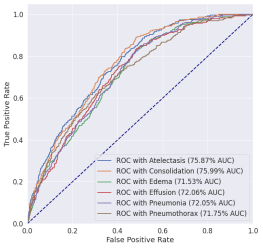
(a) Test Atelectasis



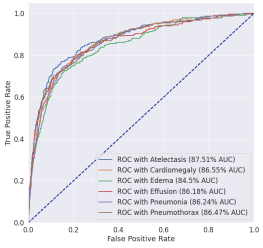
(b) Generalization Atelectasis



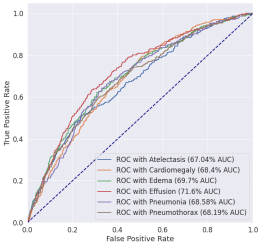
(c) Test Cardiomegaly



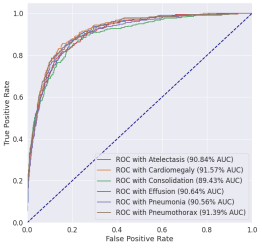
(d) Generalization Cardiomegaly



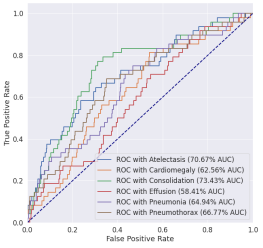
(e) Test Consolidation



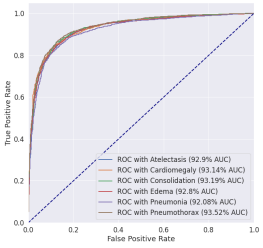
(f) Generalization Consolidation



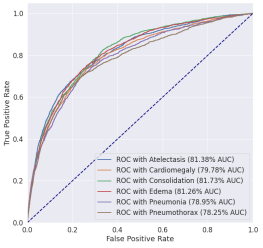
(g) Test Edema



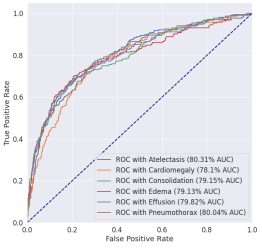
(h) Generalization Edema



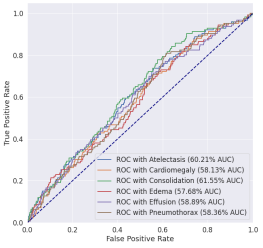
(i) Test Effusion



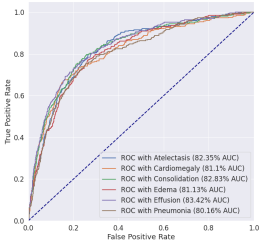
(j) Generalization Effusion



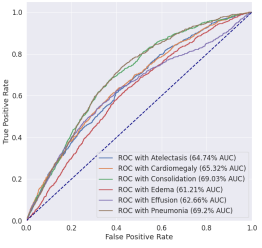
(k) Test Pneumonia



(l) Generalization Pneumonia

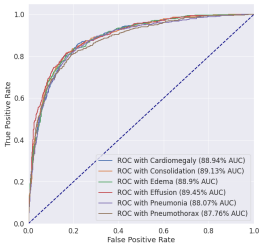


(m) Test Pneumothorax

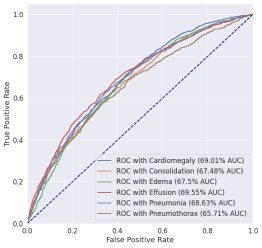


(n) Generalization Pneumothorax

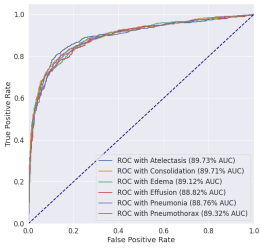
Figure 10.19: ROC curves of test performance (source) (CHEX→CHEX) and test performance (target) (CHEX→NIH) for each of the 6 pathologies when learned with the 6 other pathologies.



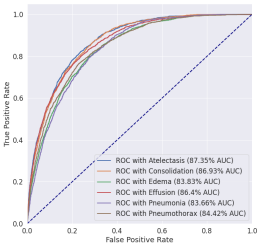
(a) Test Atelectasis



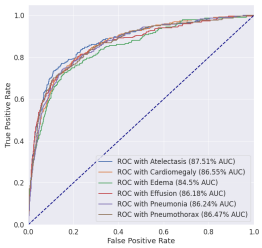
(b) Generalization Atelectasis



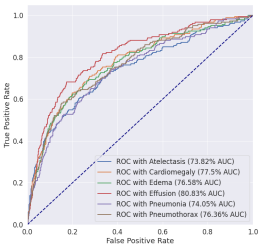
(c) Test Cardiomegaly



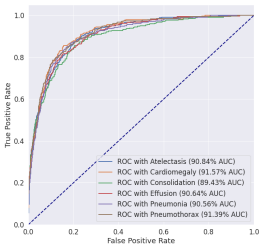
(d) Generalization Cardiomegaly



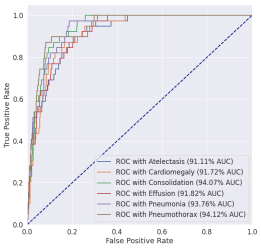
(e) Test Consolidation



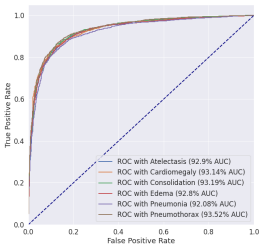
(f) Generalization Consolidation



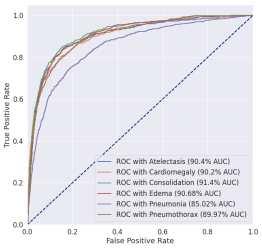
(g) Test Edema



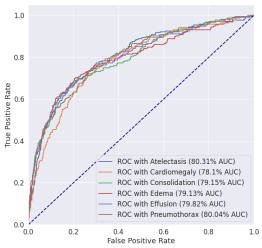
(h) Generalization Edema



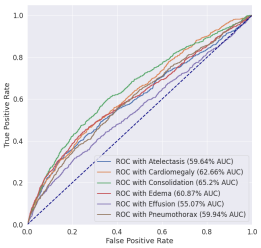
(i) Test Effusion



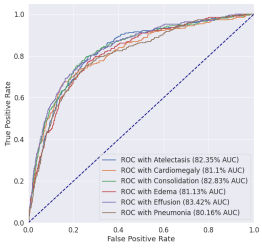
(j) Generalization Effusion



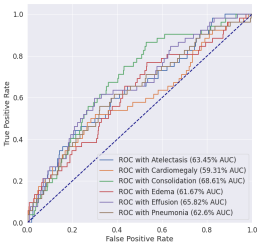
(k) Test Pneumonia



(l) Generalization Pneumonia

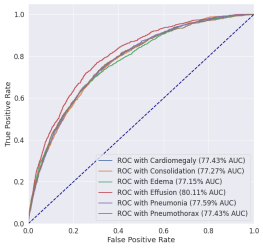


(m) Test Pneumothorax

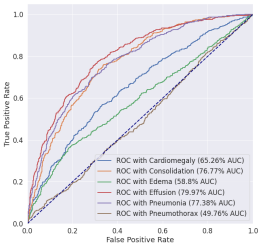


(n) Generalization Pneumothorax

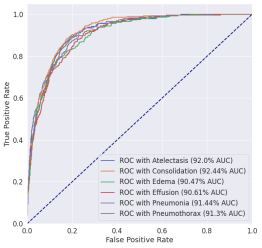
Figure 10.20: ROC curves of test performance (source) (CHEX→CHEX) and test performance (target) (CHEX→PC) for each of the 6 pathologies when learned with the 6 other pathologies.



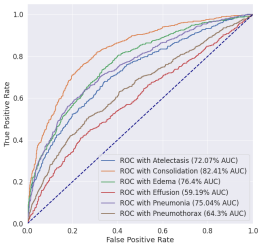
(a) Test Atelectasis



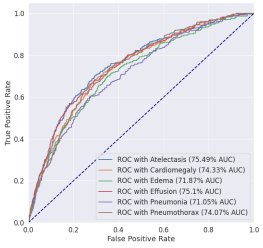
(b) Generalization Atelectasis



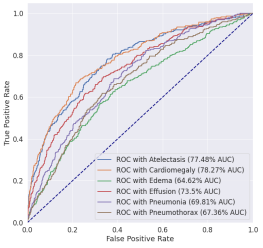
(c) Test Cardiomegaly



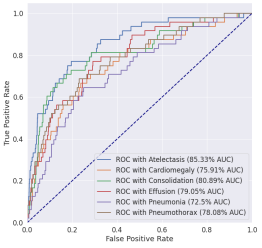
(d) Generalization Cardiomegaly



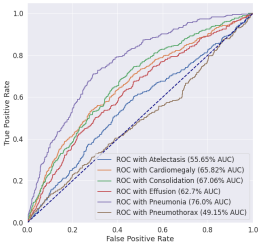
(e) Test Consolidation



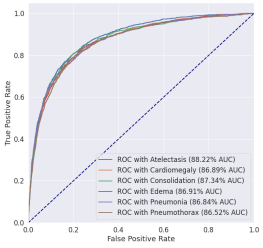
(f) Generalization Consolidation



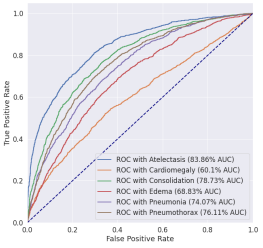
(g) Test Edema



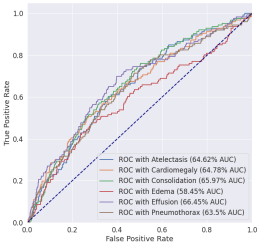
(h) Generalization Edema



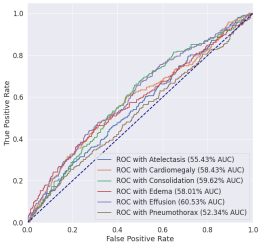
(i) Test Effusion



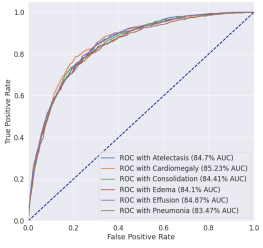
(j) Generalization Effusion



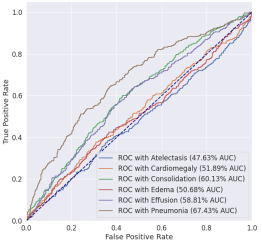
(k) Test Pneumonia



(l) Generalization Pneumonia

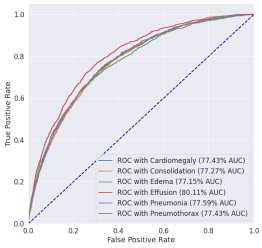


(m) Test Pneumothorax

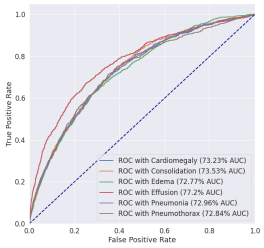


(n) Generalization Pneumothorax

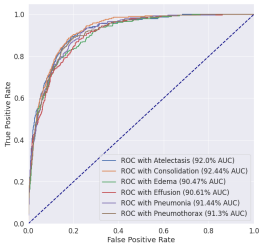
Figure 10.21: ROC curves of test performance (source) (NIH→NIH) and test performance (target) (NIH→CHEX) for each of the 6 pathologies when learned with the 6 other pathologies.



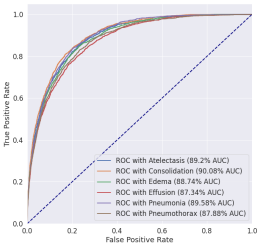
(a) Test Atelectasis



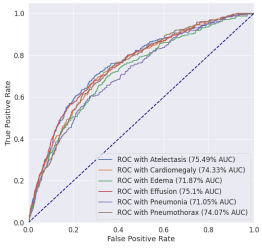
(b) Generalization Atelectasis



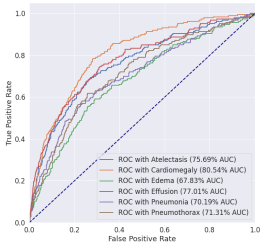
(c) Test Cardiomegaly



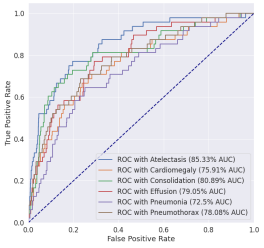
(d) Generalization Cardiomegaly



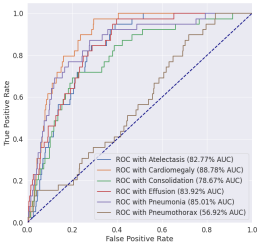
(e) Test Consolidation



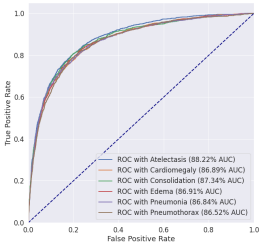
(f) Generalization Consolidation



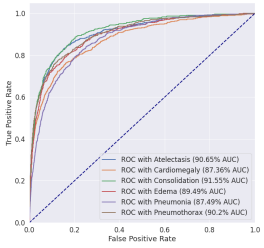
(g) Test Edema



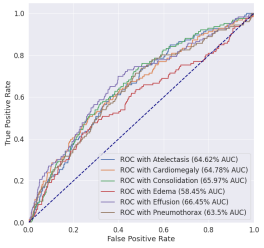
(h) Generalization Edema



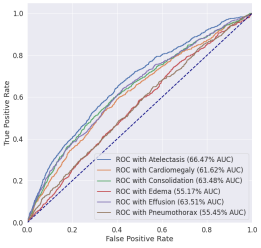
(i) Test Effusion



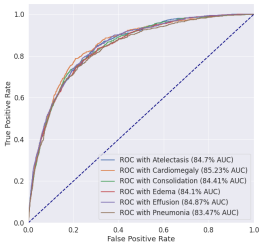
(j) Generalization Effusion



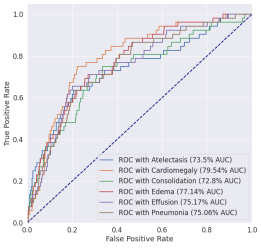
(k) Test Pneumonia



(l) Generalization Pneumonia

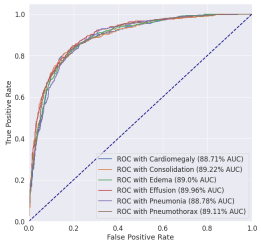


(m) Test Pneumothorax

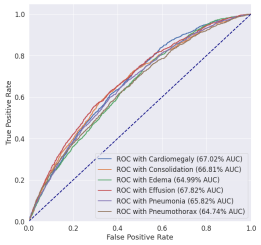


(n) Generalization Pneumothorax

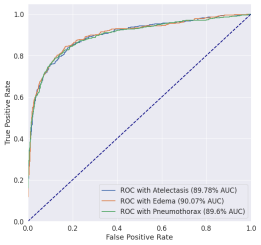
Figure 10.22: ROC curves of test performance (source) (NIH→NIH) and test performance (target) (NIH→PC) for each of the 6 pathologies when learned with the 6 other pathologies.



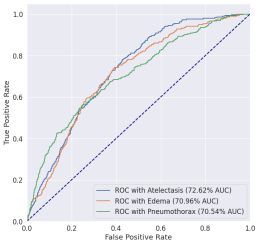
(a) Source AUC Atelectasis



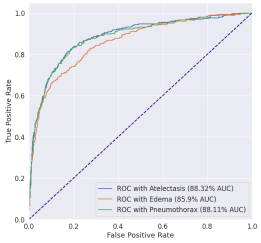
(b) Source AUC Atelectasis



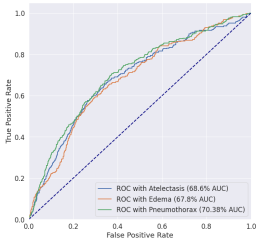
(c) Source AUC Cardiomegaly



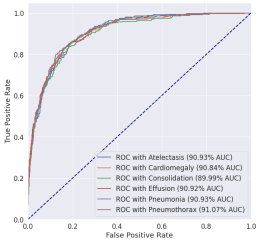
(d) Source AUC Cardiomegaly



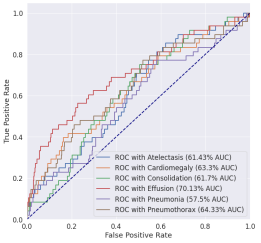
(e) Source AUC Consolidation



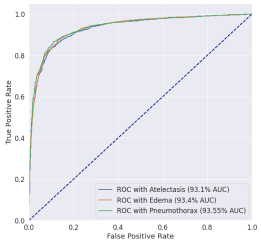
(f) Source AUC Consolidation



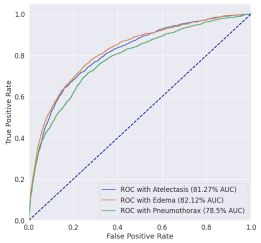
(g) Source AUC Edema



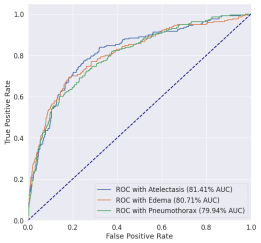
(h) Source AUC Edema



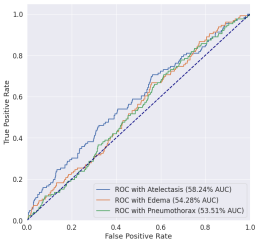
(i) Source AUC Effusion



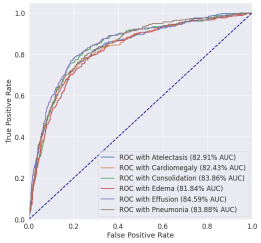
(j) Source AUC Effusion



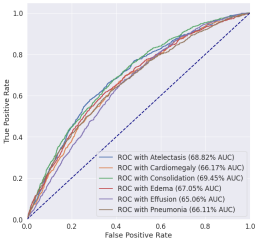
(k) Source AUC Pneumonia



(l) Source AUC Pneumonia



(m) Source AUC Pneumothorax



(n) Source AUC Pneumothorax

Figure 10.23: ROC curves of test performance (source) (CHEX→CHEX) and test performance (target) (CHEX→NIH) for each of the 6 pathologies when learned with the 6 other pathologies. The models use a DenseNet architecture

	ATE	CAR	CON	EDE	EFF	PNE	PNX
Atelectasis	0.00	-6.57	-2.51	-2.93	-2.69	-2.10	-3.77
Cardiomegaly	-3.59	0.00	-11.85	-4.89	-12.23	-8.34	-13.29
Consolidation	-2.02	-9.60	0.00	-3.17	-3.06	-0.19	-1.71
Edema	-3.28	-5.03	-5.07	0.00	-6.07	-2.48	-8.22
Effusion	-1.31	-7.75	-2.18	-5.79	0.00	-4.35	-1.89
Pneumonia	-2.18	-5.62	-3.48	-3.45	-5.38	0.00	-4.57
Pneumothorax	-6.61	-9.63	-2.75	-6.24	-3.27	-2.17	0.00

Table 10.27: Change of AUC performance on CheXpert dataset when fine-tuning the whole model using CheXpert dataset.

	ATE	CAR	CON	EDE	EFF	PNE	PNX
Atelectasis	0.00	-13.77	5.23	-0.85	-2.29	-1.99	-5.67
Cardiomegaly	-8.85	0.00	3.88	-15.87	-15.27	-0.47	-0.72
Consolidation	-3.86	-9.25	0.00	9.06	-4.52	1.04	7.47
Edema	-3.46	-7.44	2.98	0.00	-12.61	1.79	-2.07
Effusion	-0.31	-13.33	4.11	-0.25	0.00	1.40	3.59
Pneumonia	-0.65	-8.05	-2.62	-0.84	-8.79	0.00	-7.50
Pneumothorax	-4.04	-15.42	-7.95	-5.96	-8.18	5.46	0.00

Table 10.28: Change of AUC performance on NIH dataset when fine-tuning the whole model using CheXpert dataset.

10.4.3 Appendix C: Fine-tuning and encoder-freezing

In tables 10.27 and 10.28 we pre-train a model on an auxiliary pathology (rows) then fine-tuned all the model on the main pathology. Pre-training and fine-tuning are both done using the source dataset CheXpert. We then evaluate the performance of the main pathology on the CheXpert dataset in table 10.27 and on the NIH dataset in table 10.28.

In tables 10.29 and 10.30 we pre-train a model on an auxiliary pathology (rows) then fine-tuned on the decoder of each model on the main pathology, while the weights of the encoder are frozen. Pre-training and fine-tuning are both done using the source dataset CheXpert. We then evaluate the performance of the main pathology on the CheXpert dataset in table 10.29 and on the NIH dataset in table 10.30.

	ATE	CAR	CON	EDE	EFF	PNE	PNX
Atelectasis	0.00	-0.03	-0.28	-0.11	-0.13	-0.21	0.02
Cardiomegaly	0.06	0.00	-0.09	0.08	-0.62	0.15	-0.29
Consolidation	-0.35	-0.75	0.00	-0.04	-0.24	0.36	-0.29
Edema	-0.43	-0.13	0.38	0.00	-0.76	0.26	-0.89
Effusion	0.17	-0.45	-0.11	-0.10	0.00	0.12	-0.05
Pneumonia	-0.19	-0.32	0.13	-0.34	-0.35	0.00	-0.11
Pneumothorax	-0.96	-1.27	0.00	-0.82	-0.79	0.37	0.00

Table 10.29: Change of AUC performance on CheXpert dataset when fine-tuning only the decoder using CheXpert dataset.

	ATE	CAR	CON	EDE	EFF	PNE	PNX
Atelectasis	0.00	-5.07	0.38	0.17	-1.59	1.18	-1.42
Cardiomegaly	-0.21	0.00	-0.17	-0.55	-1.84	1.25	-0.49
Consolidation	-0.36	-4.34	0.00	-0.13	0.39	0.66	0.26
Edema	-0.22	-6.25	0.18	0.00	-0.83	0.79	1.30
Effusion	-0.03	-7.69	0.45	-1.64	0.00	1.02	-1.16
Pneumonia	-0.07	-1.61	-0.36	0.85	-0.09	0.00	-0.23
Pneumothorax	-0.66	-6.25	-0.56	-0.59	-0.17	0.38	0.00

Table 10.30: Change of AUC performance on NIH dataset when fine-tuning only the decoder using CheXpert dataset.

10.4.4 Appendix D: CKA patterns

[KNL+19] proposed a novel metric, **Centered kernel alignment (CKA)** that provides a reliable quantitative measure of the similarity of neural network representation.

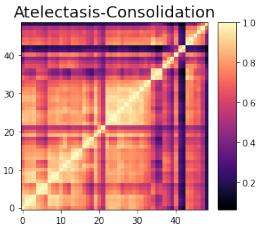
Following [NRK21], let $\mathbf{X} \in \mathbb{R}^{m \times n_1}$ and $\mathbf{Y} \in \mathbb{R}^{m \times n_2}$ the matrix representations of two layers, one with n_1 neurons and another n_2 neurons, to the same set of m examples. Each element of the $m \times m$ Gram matrices $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^\mathsf{T}$ and $\boldsymbol{L} = \boldsymbol{Y}\boldsymbol{Y}^\mathsf{T}$ reflects the similarities between a pair of examples according to the representations contained in $\boldsymbol{X}$ or $\boldsymbol{Y}$. Let $\boldsymbol{H} = \boldsymbol{I}_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\mathsf{T}$ be the centering matrix. Then $\boldsymbol{K}' = \boldsymbol{H}\boldsymbol{K}\boldsymbol{H}$ and $\boldsymbol{L}' = \boldsymbol{H}\boldsymbol{L}\boldsymbol{H}$ reflect the similarity matrices with their column and row means subtracted.

HSIC is defined as the similarity of these centered similarity matrices by reshaping them to vectors and taking the dot product between these vectors, $\text{HSIC}(\boldsymbol{K}, \boldsymbol{L}) = \frac{\text{vec}(\boldsymbol{K}')\text{vec}(\boldsymbol{L}')}{(m-1)^2}$. HSIC is invariant to orthogonal transformations of the representations and to permutation of neurons, but it is not invariant to scaling of the original representations. CKA further normalizes HSIC to produce a similarity index between 0 and 1 that is invariant to isotropic scaling,

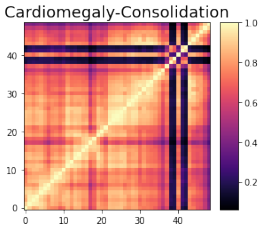
$$\text{CKA}(\boldsymbol{K}, \boldsymbol{L}) = \frac{\text{HSIC}(\boldsymbol{K}, \boldsymbol{L})}{\sqrt{\text{HSIC}(\boldsymbol{K}, \boldsymbol{K})\text{HSIC}(\boldsymbol{L}, \boldsymbol{L})}}.$$

[KNL+19] showed that linear CKA between layers of architecturally identical networks, trained from different initialization, reliably identifies architecturally corresponding layers. We implement linear CKA following the mini-batch split proposed by [NRK21].

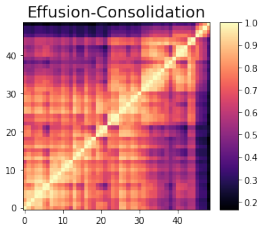
We show in figure 10.24, for each of our seven pathologies, the CKA of the combinations that lead to the best and the worst generalization performance when the models are trained on the CheXpert dataset.



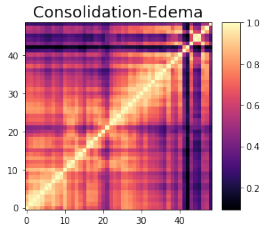
(a) Best Generalization pair for Atelectasis



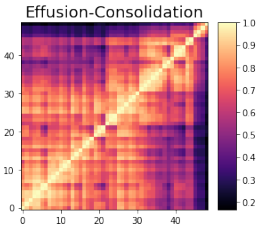
(b) Best Generalization pair for Cardiomegaly



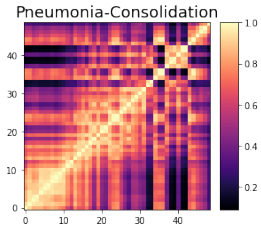
(c) Best Generalization pair for Consolidation



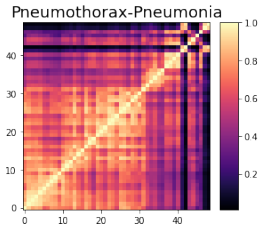
(d) Best Generalization pair for Edema



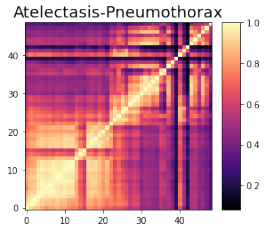
(e) Best Generalization pair for Effusion



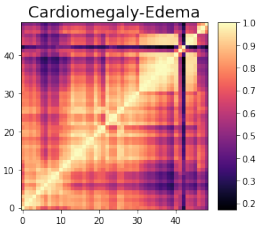
(f) Best Generalization pair for Pneumonia



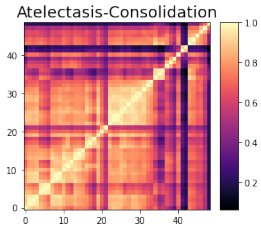
(g) Best Generalization pair for Pneumothorax



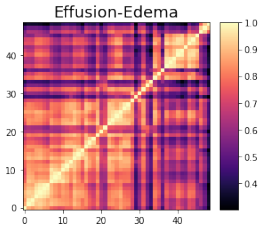
(h) Worst Generalization pair for Atelectasis



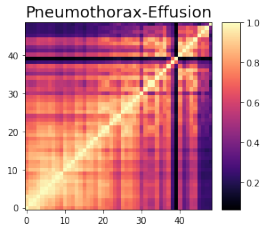
(i) Worst Generalization pair for Cardiomegaly



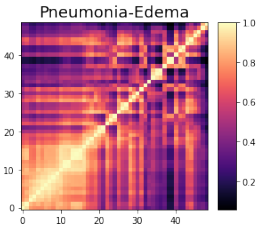
(j) Worst Generalization pair for Consolidation



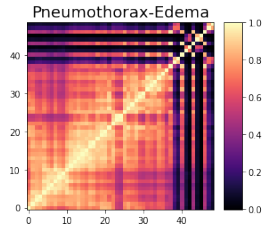
(k) Worst Generalization pair for Edema



(l) Worst Generalization pair for Effusion



(m) Worst Generalization pair for Pneumonia



(n) Worst Generalization pair for Pneumothorax

Figure 10.24: Layer Similarity for combination of models trained on CheXpert dataset

10.5 Data-driven Simulation and Optimization for Covid-19 Exit Strategies.

10.5.1 Decay functions

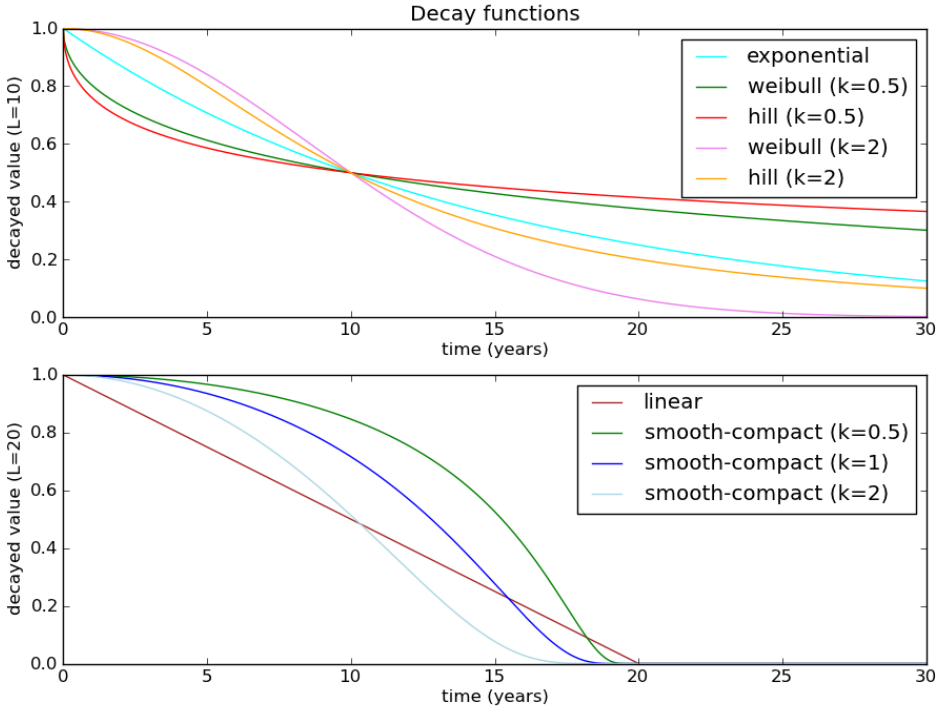


Figure 10.25: Various decay functions with L=10 and L=20

10.5.2 Extended evaluation of DN-SEIR to more countries

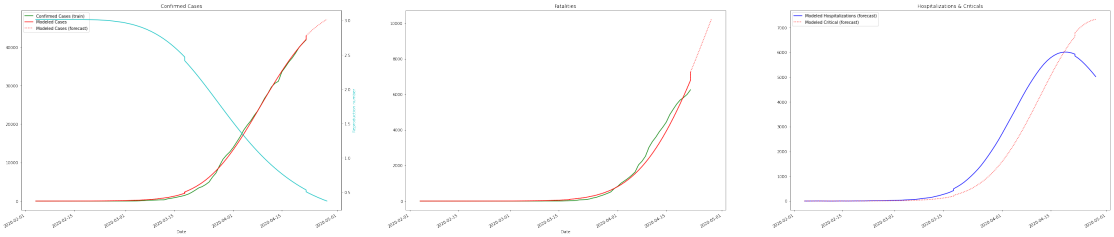


Figure 10.26: Predicted cases, hospitalizations, critical and deaths of our fitted model for Belgium.

Country	SEIR deaths	DN-SEIR deaths	True deaths
Belgium	2,180	7,049 [6,996-7,112]	7,501
France	30,382	25,493 [25,314-25,695]	24,087
Germany	48,724	7,293 [7,242-7,353]	6,115
Greece	107	107 [106-108]	139
Italy	230,989	22,476 [22,348-22,623]	27,682
Latvia	32	32 [32-33]	15
Luxembourg	78	78 [78-79]	89
Netherlands	3,364	3,348 [3,317-3,384]	4,711
Spain	81,347	19,449 [19,338-19,577]	24,543
Switzerland	1,391	1,390 [1,383-1,399]	1,716
Brazil	2.358	2,359 [2,312-2,414]	5,466
Cameroon	30	30 [29-31]	61
Canada	1,106	2,047 [2,011-2,088]	2,996
Japan	363	505 [498-512]	415
United Kingdom	22,703	19,882 [19,659-20,086]	26,771

Table 10.31: Total deaths on April 29th as predicted by the DN-SEIR model and the actual deahs. We compare the numbers with the predicted one by a SEIR model using the Reproduction rate of each country on February 15th.

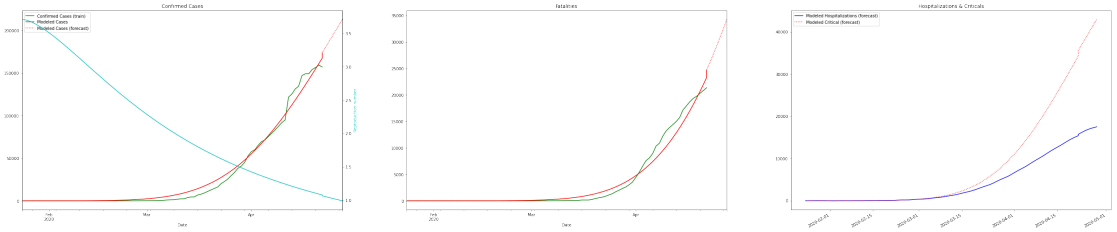


Figure 10.27: Predicted cases, hospitalizations, critical and deaths of our fitted model for France.

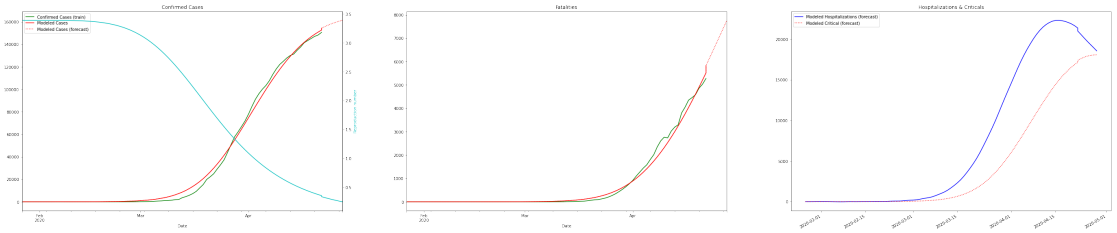


Figure 10.28: Predicted cases, hospitalizations, critical and deaths of our fitted model for Germany.

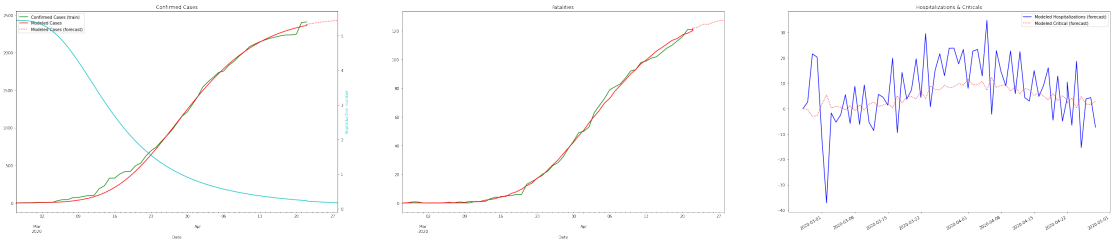


Figure 10.29: Predicted cases, hospitalizations, critical and deaths of our fitted model for Greece.

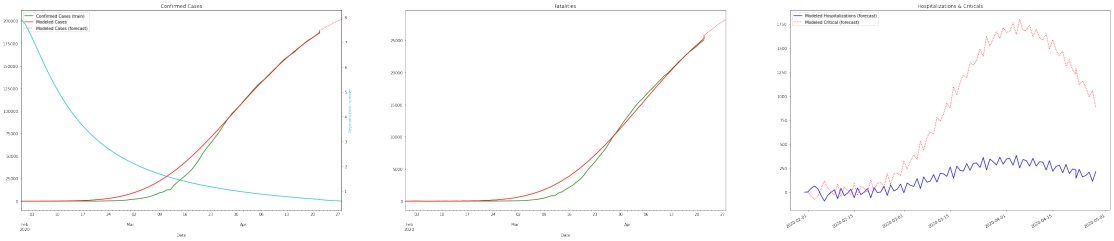


Figure 10.30: Predicted cases, hospitalizations, critical and deaths of our fitted model for Italy.

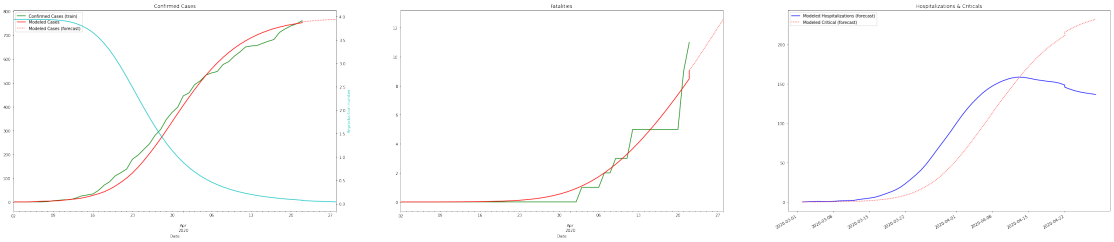


Figure 10.31: Predicted cases, hospitalizations, critical and deaths of our fitted model for Latvia.

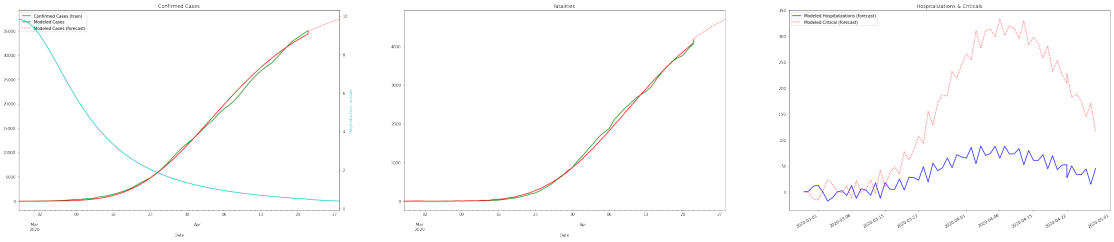


Figure 10.32: Predicted cases, hospitalizations, critical and deaths of our fitted model for Netherlands.

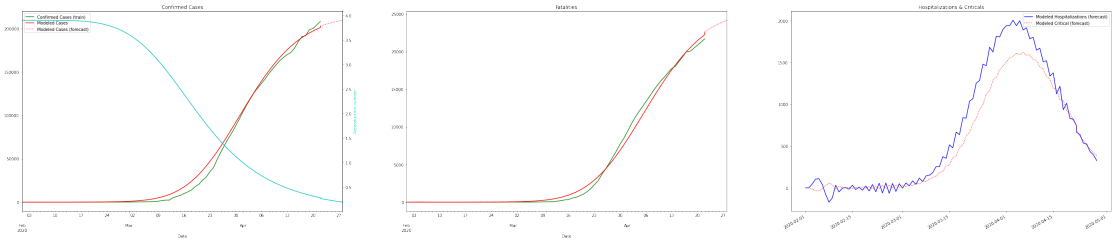


Figure 10.33: Predicted cases, hospitalizations, critical and deaths of our fitted model for Spain.

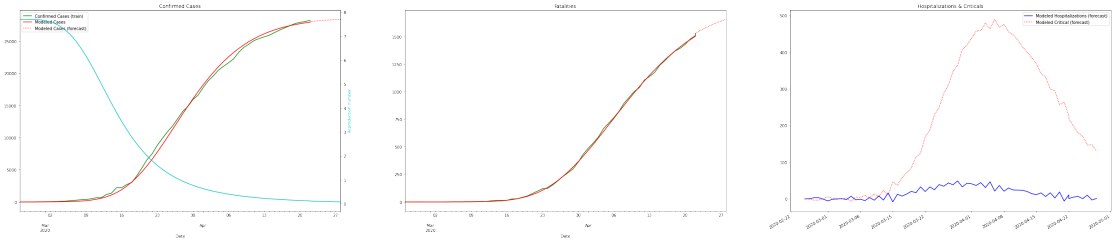


Figure 10.34: Predicted cases, hospitalizations, critical and deaths of our fitted model for Switzerland.

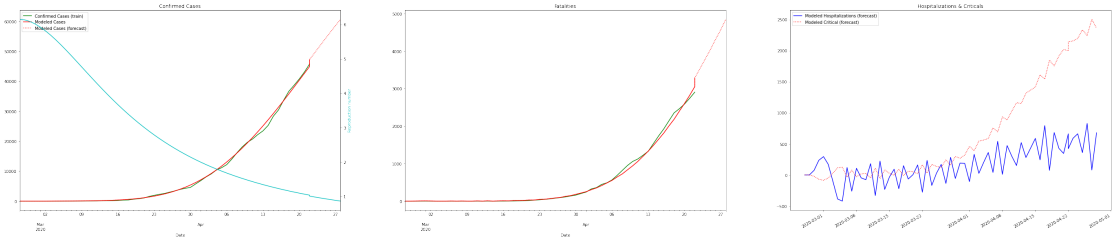


Figure 10.35: Predicted cases, hospitalizations, critical and deaths of our fitted model for Brazil.

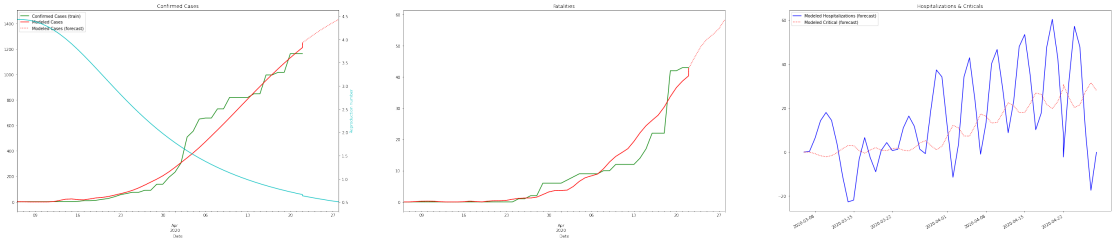


Figure 10.36: Predicted cases, hospitalizations, critical and deaths of our fitted model for Cameroon.

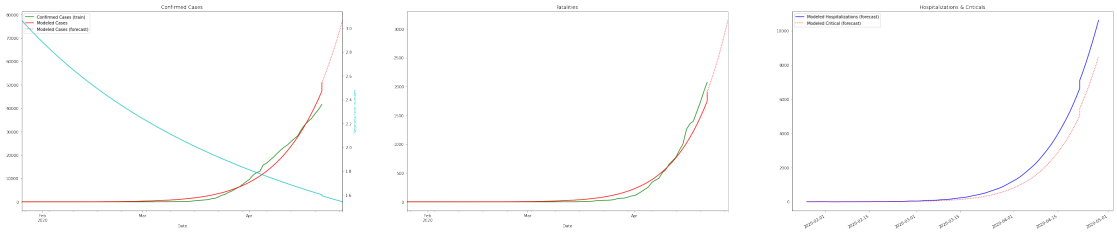


Figure 10.37: Predicted cases, hospitalizations, critical and deaths of our fitted model for Canada.

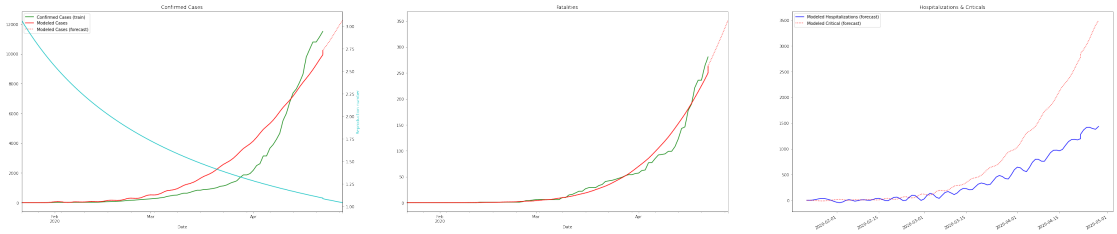


Figure 10.38: Predicted cases, hospitalizations, critical and deaths of our fitted model for Japan.

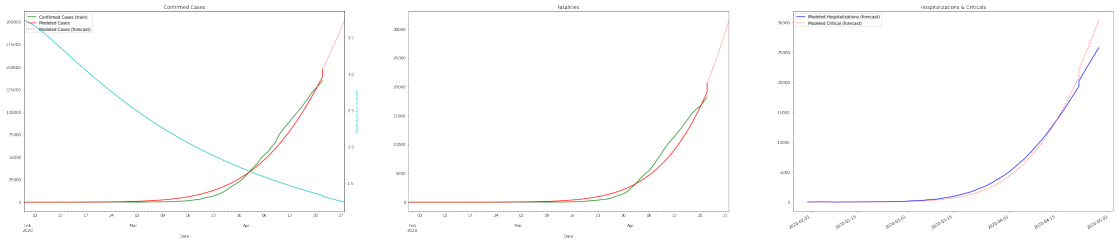


Figure 10.39: Predicted cases, hospitalizations, critical and deaths of our fitted model for United Kingdom.

List of publications and tools

Papers included in the dissertation

- Salah Ghamizi et al. Search-based adversarial testing and improvement of constrained credit scoring systems. In *Proc. of ESEC/FSE '20*, pages 1089–1100, 2020
- Salah Ghamizi et al. Adversarial robustness in multi-task learning: promises and illusions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36 of number 1, pages 697–705, 2022
- Salah Ghamizi et al. Towards generalizable machine learning for chest x-ray diagnosis with multi-task learning, 2022
- Salah Ghamizi et al. Data-driven simulation and optimization for covid-19 exit strategies. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3434–3442, 2020

Papers not included in the dissertation

- Salah Ghamizi et al. Automated search for configurations of convolutional neural network architectures. In *Proceedings of the 23rd International Systems and Software Product Line Conference-Volume A*, pages 119–130, 2019
- Salah Ghamizi et al. Pandemic simulation and forecasting of exit strategies: Convergence of machine learning and epidemiologicalmodels. Technical report, University of Luxembourg, 2020
- Salah Ghamizi et al. Featurenet: diversity-driven generation of deep learning models. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, pages 41–44, 2020
- Salah Ghamizi et al. Evasion attack steganography: turning vulnerability of machine learning to adversarial attacks into a real-world application. In *Proceedings of International Conference on Computer Vision 2021 - AROW*, 2021
- Salah Ghamizi et al. Requirements and threat models of adversarial attacks and robustness of chest x-ray classification, 2021
- Thibault Simonetto et al. A unified framework for adversarial attack and defense in constrained feature space. *arXiv preprint arXiv:2112.01156*, 2021

Tools related to the dissertation

- FeatureNet: A Neural Architecture Search approach with Feature Models
  - `https://github.com/yamizi/FeatureNet`
- CoEva2: A search-based evasion attack that generates valid adversarial examples (satisfying the domain constraints).
  - `https://github.com/thibaultsmnt/coeva2`
- SATA EAST: A new steganography and watermarking technique based on multi-label targeted evasion attacks.
  - `https://github.com/yamizi/Adversarial-Embedding`

List of figures

List of tables

Bibliography

[ACW18] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, 2018. arXiv: 1802.00420 [cs.LG] (cited on page 58).

[AIA+13] Shaukat Ali, Muhammad Zohaib Z. Iqbal, Andrea Arcuri, and Lionel C. Briand. Generating test data from OCL constraints with search techniques. *IEEE Trans. Software Eng.*, 39(10):1376–1402, 2013. DOI: 10.1109/TSE.2013.17. URL: https://doi.org/10.1109/TSE.2013.17 (cited on pages 11, 30).

[AM10] JEONGYOUN AHN and J. S. MARRON. The maximal data piling direction for discrimination. *Biometrika*, 97(1):254–259, 2010. ISSN: 00063444, 14643510. URL: http://www.jstor.org/stable/27798914 (visited on 05/11/2022) (cited on page 71).

[AM18] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: a survey. *IEEE Access*, 6:14410–14430, 2018 (cited on pages 11, 29).

[AMG+20] Sina F Ardabili, Amir Mosavi, Pedram Ghamisi, Filip Ferdinand, Annamaria R Varkonyi-Koczy, Uwe Reuter, Timon Rabczuk, and Peter M Atkinson. Covid-19 outbreak prediction with machine learning. *Available at SSRN 3580188*, 2020 (cited on page 26).

[ASC+18] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani Srivastava. Genattack: practical black-box attacks with gradient-free optimization, 2018. arXiv: 1805.11090 [cs.LG] (cited on page 24).

[ASE+18] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018 (cited on page 8).

[BAE+20] Keno K. Bressem, Lisa C. Adams, Christoph Erxleben, Bernd Hamm, Stefan M. Niehues, and Janis L. Vahldiek. Comparing different deep learning architectures for classification of chest radiographs. *Scientific Reports*, 10(1), August 2020. ISSN: 2045-2322. DOI: 10.1038/s41598-020-70479-z. URL: http://dx.doi.org/10.1038/s41598-020-70479-z (cited on page 89).

[BBa20] Emma Beede, Elizabeth Baylor, and al. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020 (cited on pages 12, 84).

[BCG18] Lindsay P. Busby, Jesse L. Courtier, and Christine M. Glastonbury. Bias in radiology: the how and why of misses and misinterpretations. *RadioGraphics*, 38(1):236–247, 2018. DOI: 10.1148/rg.2018170107. eprint: https://doi.org/10.1148/rg.2018170107. URL: https://doi.org/10.1148/rg.2018170107. PMID: 29194009 (cited on page 12).

[BCM+13] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim ***v****Srndić*, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8190 LNAI of number PART 3, pages 387–402, August 2013. ISBN: 9783642409936. DOI: 10.1007/978-3-642-40994-3_25. arXiv: 1708.06131. URL: http://arxiv.org/abs/1708.06131%20http://dx.doi.org/10.1007/978-3-642-40994-3%7B%5C_%7D25 (cited on pages 7, 20, 41, 54).

[BNG+19] Ivo M. Baltruschat, Hannes Nickisch, Michael Grass, Tobias Knopp, and Axel Saalbach. Comparison of deep learning approaches for multi-label chest x-ray classification, 2019. arXiv: 1803.02315 [cs.CV] (cited on pages 12, 89).

[BNL12] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012 (cited on page 20).

[BPA17] Michael A Bruno, Jonelle Petscavage-Thomas, and Hani H Abujudeh. Communicating uncertainty in the radiology report. *American Journal of Roentgenology*, 209(5):1006–1008, 2017 (cited on page 84).

[BPS+20] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vayá. Padchest: a large chest x-ray image dataset with multi-label annotated reports. *Medical Image Analysis*, 66:101797, December 2020. ISSN: 1361-8415. DOI: 10.1016/j.media.2020.101797. URL: http://dx.doi.org/10.1016/j.media.2020.101797 (cited on page 87).

[BR18] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, December 2018. ISSN: 00313203. DOI: 10.1016/j.patcog.2018.07.023. arXiv: 1712.03141. URL: http://arxiv.org/abs/1712.03141%20http://dx.doi.org/10.1016/j.patcog.2018.07.023 (cited on pages 11, 29).

[BTO+19] Felix J. S. Bragman, Ryutaro Tanno, Sebastien Ourselin, Daniel C. Alexander, and M. Jorge Cardoso. Stochastic filter groups for multi-task cnns: learning specialist and generalist convolution kernels, 2019. DOI: 10.48550/ARXIV.1908.09597. URL: https://arxiv.org/abs/1908.09597 (cited on page 21).

[CAS+20] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020 (cited on page 70).

[CBL+18] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: gradient normalization for adaptive loss balancing in deep multitask networks, 2018. arXiv: 1711.02257 [cs.CV] (cited on pages 22, 55, 56, 64, 65).

[CDB+19] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles, 2019. arXiv: 1903.06864 [cs.CV] (cited on page 21).

[CH20a] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020 (cited on pages 55, 58, 59, 64, 157).

[CH20b] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020 (cited on page 80).

[CHB+20] Joseph Paul Cohen, Mohammad Hashir, Rupert Brooks, and Hadrien Bertrand. On the limits of cross-domain generalization in automated x-ray prediction, 2020. arXiv: 2002.02497 [eess.IV] (cited on pages 12, 25, 84, 86, 88, 89, 95, 156, 166).

[CLC+20] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: from self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 699–708, 2020 (cited on page 20).

[CNL11] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011 (cited on page 81).

[Com20] European Commission. White paper on artificial intelligence: a european approach to excellence and trust. *Com (2020) 65 Final*, 2020 (cited on page 1).

[CRS+19] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019 (cited on page 20).

[CVB+21] Joseph Paul Cohen, Joseph D. Viviano, Paul Bertin, Paul Morrison, Parsa Torabian, Matteo Guarrera, Matthew P Lungren, Akshay Chaudhari, Rupert Brooks, Mohammad Hashir, and Hadrien Bertrand. Torchxrayvision: a library of chest x-ray datasets and models, 2021. arXiv: 2111.00595 [eess.IV] (cited on page 88).

[CW17] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. *Proceedings - IEEE Symposium on Security and Privacy*:39–57, 2017. ISSN: 10816011. DOI: 10.1109/SP.2017.49. arXiv: arXiv:1608.04644v2 (cited on page 37).

[CWG20] Daniel C Castro, Ian Walker, and Ben Glocker. Causality matters in medical imaging. *Nature Communications*, 11(1):1–10, 2020 (cited on pages 6–8).

[CXC+19] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC conference on*

computer and communications security, pages 2267–2281, 2019 (cited on page 70).

[Des12] Jean-Antoine Desideri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5):313–318, 2012. ISSN: 1631-073X. DOI: https://doi.org/10.1016/j.crma.2012.03.014. URL: https://www.sciencedirect.com/science/article/pii/S1631073X12000738 (cited on page 73).

[DGS+22] Salijona Dyrmishi, Salah Ghamizi, Thibault Simonetto, Yves Le Traon, and Maxime Cordy. On the empirical effectiveness of unrealistic adversarial hardening against realistic adversarial attacks, 2022. arXiv: 2202.03277 [cs.LG] (cited on page 20).

[DH20] Alexander D'Amour and Katherine et al. Heller. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020 (cited on pages 12, 84).

[DLP+18] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018 (cited on page 20).

[dMB16] Vinícius Veloso de Melo and Wolfgang Banzhaf. Improving Logistic Regression Classification of Credit Approval with Features Constructed by Kaizen Programming. en. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion - GECCO '16 Companion*, pages 61–62, Denver, Colorado, USA. ACM Press, 2016. ISBN: 978-1-4503-4323-7. DOI: 10.1145/2908961.2908963. URL: http://dl.acm.org/citation.cfm?doid=2908961.2908963 (visited on 01/16/2020) (cited on pages 10, 23, 29).

[DPA+02] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: nsga-ii. In volume 6 of number 2, pages 182–197, April 2002. DOI: 10.1109/4235.996017 (cited on page 113).

[DR19] Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12387–12396, 2019 (cited on page 22).

[DSO07] Kalyanmoy Deb, Karthik Sindhya, and Tatsuya Okabe. Self-adaptive simulated binary crossover for real-parameter optimization. GECCO '07:1187–1194, 2007. DOI: 10.1145/1276958.1277190. URL: https://doi.org/10.1145/1276958.1277190 (cited on page 43).

[DSS+21] Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. Robustness in deep learning for computer vision: mind the gap? *arXiv preprint arXiv:2112.00639*, 2021 (cited on page 6).

[FKB18] Samuel Finlayson, Isaac Kohane, and Andrew Beam. Adversarial attacks against medical deep learning systems, April 2018 (cited on pages 13, 25).

[FLH+19] Hongwei Feng, Shuang Li, Dianyuan He, and Jun Feng. A novel feature selection approach based on multiple filters and new separable degree index for credit scoring. en. In *Proceedings of the ACM Turing Celebration Conference - China on - ACM TURC '19*, pages 1–5, Chengdu, China. ACM Press, 2019. ISBN: 978-1-4503-7158-2. DOI: 10.1145/3321408.3323928. URL: http://dl.acm.org/citation.cfm?doid=3321408.3323928 (visited on 01/16/2020) (cited on pages 10, 23, 29).

[FS14] Thomas Ferkol and Dean Schraufnagel. The global burden of respiratory disease. *Annals of the American Thoracic Society*, 11(3):404–406, 2014 (cited on page 99).

[FS19] Jason Furman and Robert Seamans. Ai and the economy. *Innovation policy and the economy*, 19(1):161–191, 2019 (cited on page 1).

[GCD09] B. J. Garvin, M. B. Cohen, and M. B. Dwyer. An improved meta-heuristic search for constrained interaction testing. In *2009 1st International Symposium on Search Based Software Engineering*, pages 13–22, 2009 (cited on page 24).

[GCG+20a] Salah Ghamizi, Maxime Cordy, Martin Gubri, Mike Papadakis, Andrey Boystov, Yves Le Traon, and Anne Goujon. Search-based adversarial testing and improvement of constrained credit scoring systems. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, pages 1089–1100, Virtual Event, USA. Association for Computing Machinery, 2020. ISBN: 9781450370431. DOI: 10.1145/3368089.3409739. URL: https://doi.org/10.1145/3368089.3409739 (cited on page 20).

[GCG+20b] Salah Ghamizi, Maxime Cordy, Martin Gubri, Mike Papadakis, Andrey Boystov, Yves Le Traon, and Anne Goujon. Search-based adversarial testing and improvement of constrained credit scoring systems. In *Proc. of ESEC/FSE '20*, pages 1089–1100, 2020 (cited on pages 8, iii).

[GCP+19a] Salah Ghamizi, Maxime Cordy, Mike Papadakis, and Yves Le Traon. Adversarial embedding: a robust and elusive steganography and watermarking technique. *arXiv preprint arXiv:1912.01487*, 2019 (cited on page 54).

[GCP+19b] Salah Ghamizi, Maxime Cordy, Mike Papadakis, and Yves Le Traon. Automated search for configurations of convolutional neural network architectures. In *Proceedings of the 23rd International Systems and Software Product Line Conference-Volume A*, pages 119–130, 2019 (cited on page iii).

[GCP+20] Salah Ghamizi, Maxime Cordy, Mike Papadakis, and Yves Le Traon. Featurenet: diversity-driven generation of deep learning models. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, pages 41–44, 2020 (cited on page iii).

[GCP+21a] Salah Ghamizi, Maxime Cordy, Mike Papadakis, and Yves Le Traon. Evasion attack steganography: turning vulnerability of machine learning to adversarial attacks into a real-world application. In *Proceedings of International Conference on Computer Vision 2021 - AROW*, 2021 (cited on page iii).

[GCP+21b] Salah Ghamizi, Maxime Cordy, Mike Papadakis, and Yves Le Traon. Requirements and threat models of adversarial attacks and robustness of chest x-ray classification, 2021 (cited on page iii).

[GCP+21c] Salah Ghamizi, Maxime Cordy, Mike Papadakis, and Yves Le Traon. Adversarial robustness in multi-task learning: promises and illusions, 2021. DOI: 10.48550/ARXIV.2110.15053. URL: https://arxiv.org/abs/2110.15053 (cited on page 8).

[GCP+21d] Salah Ghamizi, Maxime Cordy, Mike Papadakis, and Yves Le Traon. Adversarial robustness in multi-task learning: promises and illusions. *arXiv preprint arXiv:2110.15053*, 2021 (cited on page 89).

[GCP+22] Salah Ghamizi, Maxime Cordy, Mike Papadakis, and Yves Le Traon. Adversarial robustness in multi-task learning: promises and illusions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36 of number 1, pages 697–705, 2022 (cited on page iii).

[GGT+22] Salah Ghamizi, Beatriz Garcia Santa Cruz, Paul Temple, Maxime Cordy, Gilles Perrouin, Mike Papadakis, and Yves Le Traon. Towards generalizable machine learning for chest x-ray diagnosis with multi-task learning, 2022 (cited on page iii).

[GLS+19] Ting Gong, Tyler Lee, Cory Stephenson, Venkata Renduchintala, Suchismita Padhy, Anthony Ndirango, Gokce Keskin, and Oguz H Elibol. A comparison of loss weighting strategies for multi task learning in deep neural networks. *IEEE Access*, 7:141627–141632, 2019 (cited on page 22).

[GMZ+19] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3205–3214, 2019 (cited on page 21).

[GRC+20a] Salah Ghamizi, Renaud Rwemalika, Maxime Cordy, Yves Le Traon, and Mike Papadakis. Pandemic simulation and forecasting of exit strategies: Convergence of machine learning and epidemiologicalmodels. Technical report, University of Luxembourg, 2020 (cited on page iii).

[GRC+20b] Salah Ghamizi, Renaud Rwemalika, Maxime Cordy, Lisa Veiber, Tegawendé F Bissyandé, Mike Papadakis, Jacques Klein, and Yves Le Traon. Data-driven simulation and optimization for covid-19 exit strategies. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3434–3442, 2020 (cited on page iii).

[GRL+19] Prasanth Ganesan, Sivaramakrishnan Rajaraman, Rodney Long, Behnaz Ghoraani, and Sameer Antani. Assessment of data augmentation strategies toward performance improvement of abnormality classification in chest radiographs. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 841–844. IEEE, 2019 (cited on pages 72, 156).

[GRW+21] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34, 2021 (cited on pages 20, 70).

[GSK18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018 (cited on page 72).

[GSS14a] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014 (cited on pages 7, 20, 54).

[GSS14b] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014 (cited on page 56).

[HWC+17] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: ensembles of weak defenses are not strong. In *11th {USENIX} workshop on offensive technologies ({WOOT} 17)*, 2017 (cited on pages 7, 54).

[HZC+19] Shoubo Hu, Kun Zhang, Zhitang Chen, and Laiwan Chan. Domain generalization via multidomain discriminant analysis, 2019. arXiv: 1907.11216 [stat.ML] (cited on page 21).

[HZR+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016 (cited on pages 76, 156).

[IRa19] Jeremy Irvin, Pranav Rajpurkar, and al. Chexpert: a large chest radiograph dataset with uncertainty labels and expert comparison, 2019. arXiv: 1901.07031 [cs.CV] (cited on pages 72, 75, 84, 87, 89, 95, 159).

[IWL+19] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7066–7074, 2019 (cited on pages 7, 54).

[JB11] Mark Jit and Marc Brisson. Modelling the Epidemiology of Infectious Diseases for Decision Analysis. *PharmacoEconomics*, 29(5):371–386, May 2011. ISSN: 1170-7690 (cited on pages 13, 105).

[JLZ+20] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Feature alignment and restoration for domain generalization and adaptation, 2020. arXiv: 2006.12009 [cs.CV] (cited on page 21).

[KCD+00] Linda T Kohn, Janet M Corrigan, Molla S Donaldson, et al. Errors in health care: a leading cause of death and injury, 2000 (cited on page 12).

[KGB16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016 (cited on pages 7, 20, 54).

[KGC18] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018 (cited on page 22).

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009 (cited on page 75).

[KHS22] Sara Kaviani, Ki Jin Han, and Insoo Sohn. Adversarial attacks and defenses on ai in medical imaging informatics: a survey. *Expert Systems with Applications*, 198:116815, 2022. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.116815. URL: https://www.sciencedirect.com/science/article/pii/S095741742200272X (cited on page 71).

[KLP+21] Jin Kim, Jiyoung Lee, Jungin Park, Dongbo Min, and Kwanghoon Sohn. Self-balanced learning for domain generalization. *2021 IEEE International Conference on Image Processing (ICIP)*, September 2021. DOI: 10.1109/icip42928.2021.9506516. URL: http://dx.doi.org/10.1109/ICIP42928.2021.9506516 (cited on page 21).

[KM18] Fumito Koike and Nobuo Morimoto. Supervised forecasting of the range expansion of novel non-indigenous organisms: alien pest organisms and the 2009 h1n1 flu pandemic. *Global Ecology and Biogeography*, April 2018 (cited on page 26).

[KNL+19] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited, 2019. arXiv: 1905.00414 [cs.LG] (cited on pages 90, 97, 179).

[Kok16] Iasonas Kokkinos. Ubernet: training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory, 2016. DOI: 10.48550/ARXIV.1609.02132. URL: https://arxiv.org/abs/1609.02132 (cited on page 21).

[KTJ16] Alex Kantchelian, J. Doug Tygar, and Anthony Joseph. Evasion and hardening of tree ensemble classifiers. In *International Conference on Machine Learning*, pages 2387–2396, 2016 (cited on page 23).

[LAF16] Francisco Louzada, Anderson Ara, and Guilherme B. Fernandes. Classification methods applied to credit scoring: Systematic review and overall comparison. en. *Surveys in Operations Research and Management Science*, 21(2):117–134, December 2016. ISSN: 18767354. DOI: 10.1016/j.sorms.2016.10.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S1876735416300101 (visited on 01/16/2020) (cited on page 23).

[LGW+20] Ying Liu, Albert A. Gayle, Annelies Wilder-Smith, and Joacim Rocklöv. The reproductive number of COVID-19 is higher compared to SARS coronavirus. *Journal of Travel Medicine*, 27(2):1–4, March 2020. ISSN: 1708-8305 (cited on page 109).

[LJD19] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019 (cited on page 22).

[LKD63] Gwilym S. Lodwick, Theodore E. Keats, and John P. Dorst. The coding of roentgen images for computer analysis as applied to lung cancer. *Radiology*, 81(2):185–200, 1963. DOI: 10.1148/81.2.185. eprint: https://doi.org/10.1148/81.2.185. URL: https://doi.org/10.1148/81.2.185. PMID: 14053755 (cited on page 12).

[LLK+21] Liyang Liu, Yi Li, Zhanghui Kuang, J Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In ICLR, 2021 (cited on pages 73, 162).

[LLY+20] Jiangnan Li, Jin Young Lee, Yingyuan Yang, Jinyuan Stella Sun, and Kevin Tomsovic. Conaml: constrained adversarial machine learning for cyber-physical systems. *arXiv preprint arXiv:2003.05631*, 2020 (cited on page 8).

[LSB+20] Isabelle Leang, Ganesh Sistu, Fabian Bürger, Andrei Bursuc, and Senthil Yogamani. Dynamic task weighting methods for multi-task networks in autonomous driving systems. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020 (cited on page 22).

[LZ20] Xin Li and Dongxiao Zhu. Robust detection of adversarial attacks on medical images. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1154–1158, 2020. DOI: 10.1109/ISBI45749.2020.9098628 (cited on page 13).

[McM04] Phil McMinn. Search-based software test data generation: a survey. *Softw. Test. Verification Reliab.*, 14(2):105–156, 2004. DOI: 10.1002/stvr.294. URL: https://doi.org/10.1002/stvr.294 (cited on pages 24, 41).

[MGF+17] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017 (cited on pages 7, 54).

[MGN+20] Chengzhi Mao, Amogh Gupta, Vikram Nitin, Baishakhi Ray, Shuran Song, Junfeng Yang, and Carl Vondrick. Multitask learning strengthens adversarial robustness. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12347 of *Lecture Notes in Computer Science*, pages 158–174. Springer, 2020. DOI: 10.1007/978-3-030-58536-

5_10. URL: https://doi.org/10.1007/978-3-030-58536-5%5C_10 (cited on pages 22, 54, 57, 60, 62, 64).

[MH20a] Esteban Ortiz-Ospina Max Roser Hannah Ritchie and Joe Hasell. Coronavirus pandemic (covid-19). *Our World in Data*, 2020. https://ourworldindata. (cited on page 119).

[MH20b] Gautam Raj Mode and Khaza Anuarul Hoque. Crafting adversarial examples for deep learning based prognostics (extended version). *arXiv preprint arXiv:2009.10149*, 2020 (cited on page 8).

[MMS+17a] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks:1–27, 2017. arXiv: 1706.06083. URL: http://arxiv.org/abs/1706.06083 (cited on page 37).

[MMS+17b] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017 (cited on pages 7, 20, 54, 59, 64, 156).

[MMS+17c] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017 (cited on pages 20, 70, 76, 156).

[MMW+20] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms, 2020. arXiv: 2010.07922 [cs.LG] (cited on page 21).

[MNG+21] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 110:107332, 2021. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107332. URL: https://www.sciencedirect.com/science/article/pii/S0031320320301357 (cited on pages 13, 25, 70).

[Mol19] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2019 (cited on page 113).

[MS96] Zbigniew Michalewicz and Marc Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evol. Comput.*, 4(1):1–32, 1996. DOI: 10.1162/evco.1996.4.1.1. URL: https://doi.org/10.1162/evco.1996.4.1.1 (cited on pages 24, 41).

[MSG+16] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning, 2016. DOI: 10.48550/ARXIV.1604.03539. URL: https://arxiv.org/abs/1604.03539 (cited on page 21).

[MSI+18] Olav Titus Muurlink, Peter Stephenson, Mohammad Zahirul Islam, and Andrew W Taylor-Robinson. Long-term predictors of dengue outbreaks in bangladesh: a data mining approach. *Infectious Disease Modelling*, 3:322–330, 2018 (cited on page 26).

[MTS21] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching, 2021. arXiv: 2006.07500 [cs.LG] (cited on page 25).

[NF16] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016 (cited on page 72).

[NMK+19] Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to adversarial perturbations in learning from incomplete data. *Advances in Neural Information Processing Systems*, 32, 2019 (cited on page 70).

[NRK21] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth, 2021. arXiv: 2010.15327 [cs.LG] (cited on pages 90, 179).

[NST+18] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. Adversarial robustness toolbox v1.1.1. *CoRR*, 1807.01069, 2018. URL: https://arxiv.org/pdf/1807.01069 (cited on page 37).

[Oak19] Luke Oakden-Rayner. Exploring large scale public medical image datasets, 2019. arXiv: 1907.12720 [eess.IV] (cited on page 12).

[PBB20] Eduardo H. P. Pooch, Pedro L. Ballester, and Rodrigo C. Barros. Can we trust deep learning models diagnosis? the impact of domain shift in chest radiograph classification, 2020. arXiv: 1909.01940 [eess.IV] (cited on pages 12, 25).

[PBV12] Mohamad Amin Pourhoseingholi, Ahmad Reza Baghestani, and Mohsen Vahedi. How to control confounding effects by statistical analysis. *Gastroenterology and hepatology from bed to bench*, 5(2):79, 2012 (cited on page 9).

[PCG+20] Gaurav Pandey, Poonam Chaudhary, Rajan Gupta, and Saibal Pal. Seir and regression model based covid-19 outbreak predictions in india. *arXiv preprint*, 2020 (cited on page 26).

[PLS20] Xingchao Peng, Yichen Li, and Kate Saenko. Domain2vec: domain embedding for unsupervised domain adaptation, 2020. arXiv: 2007.09257 [cs.CV] (cited on page 21).

[PMG16] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016. arXiv: 1605.07277. URL: http://arxiv.org/abs/1605.07277 (cited on pages 23, 30, 36).

[PMJ+16] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016 (cited on pages 11, 23, 29).

[PPC+20] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. Intriguing properties of adversarial ml attacks in the problem space. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1332–1349. IEEE, 2020 (cited on page 8).

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015 (cited on page 72).

[RGC+21] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34, 2021 (cited on pages 20, 70).

[RIZ+17] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. Chexnet: radiologist-level pneumonia detection on chest x-rays with deep learning, 2017. arXiv: 1711.05225 [cs.CV] (cited on page 12).

[RKA17] Pinyao Rui, K Kang, and Michael Albert. National hospital ambulatory medical care survey: 2015 emergency department summary tables. *National center for health statistics*, 2017 (cited on page 99).

[Rud19] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. en. *Nature Machine Intelligence*, 1(5):206–215, May 2019. ISSN: 2522-5839. DOI: 10.1038/s42256-019-0048-x. URL: http://www.nature.com/articles/s42256-019-0048-x (visited on 01/20/2020) (cited on page 32).

[RVB18] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018 (cited on page 54).

[SCC+20a] Nicholas Soures, David Chambers, Zachariah Carmichael, Anurag Daram, Dimpy P Shah, Kal Clark, Lloyd Potter, and Dhireesha Kudithipudi. SIRNet: Understanding Social Distancing Measures with Hybrid Neural Network Model for COVID-19 Infectious Spread. Technical report, 2020. eprint: 2004.10376 (cited on pages 14, 106).

[SCC+20b] Nicholas Soures, David Chambers, Zachariah Carmichael, Anurag Daram, Dimpy P. Shah, Kal Clark, Lloyd Potter, and Dhireesha Kudithipudi. Sirnet: understanding social distancing measures with hybrid neural network model for covid-19 infectious spread, 2020 (cited on page 26).

[SCF18] Roberto Saia, Salvatore Carta, and Gianni Fenu. A Wavelet-based Data Analysis to Credit Scoring. en. In *Proceedings of the 2nd International Conference on Digital Signal Processing - ICDSP 2018*, pages 176–180, Tokyo, Japan. ACM Press, 2018. ISBN: 978-1-4503-6402-7. DOI: 10.1145/3193025.3193039. URL: http://dl.acm.org/citation.cfm?doid=3193025.3193039 (visited on 01/16/2020) (cited on pages 10, 23, 29).

[SDG+21] Thibault Simonetto, Salijona Dyrmishi, Salah Ghamizi, Maxime Cordy, and Yves Le Traon. A unified framework for adversarial attack and defense in constrained feature space. *arXiv preprint arXiv:2112.01156*, 2021 (cited on page iii).

[SEZ+18] Alexander Sax, Bradley Emi, Amir R Zamir, Leonidas Guibas, Silvio Savarese, and Jitendra Malik. Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies. *arXiv preprint arXiv:1812.11971*, 2018 (cited on page 54).

[SK18] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.,

2018. URL: https://proceedings.neurips.cc/paper/2018/file/432aca3a1e345e339f35a30c8f65edce-Paper.pdf (cited on pages 22, 73, 87, 162).

[SK21] Nithya Sambasivan and Shivani et al. Kapania. "everyone wants to do the model work, not the data work": data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021 (cited on page 99).

[SKM+19] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: a platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019 (cited on page 54).

[SMR+08] T. Smith, N. Maire, A. Ross, M. Penny, N. Chitnis, A. Schapira, A. Studer, B. Genton, C. Lengeler, F. Tediosi, and et al. Towards a comprehensive simulation model of malaria epidemiology and control. *Parasitology*, 135(13):1507–1516, 2008 (cited on page 110).

[SNG+19] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019 (cited on page 20).

[SNV+17] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017 (cited on page 56).

[SOB+19] Carl-Johann Simon-Gabriel, Yann Ollivier, Leon Bottou, Bernhard Schölkopf, and David Lopez-Paz. First-order adversarial vulnerability of neural networks and input dimension. In *International Conference on Machine Learning*, pages 5809–5817. PMLR, 2019 (cited on pages 22, 56, 57, 67).

[SPW+20] Ryan Sheatsley, Nicolas Papernot, Michael Weisman, Gunjan Verma, and Patrick McDaniel. Adversarial examples in constrained domains. *arXiv preprint arXiv:2011.01183*, 2020 (cited on page 8).

[SST+18] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31, 2018 (cited on page 70).

[SZC+20] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020 (cited on pages 21, 22, 54, 55, 59, 64, 65, 67).

[SZS+13] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013 (cited on pages 7, 11, 29, 54).

[TDH18] Saeid Asgari Taghanaki, Arkadeep Das, and Ghassan Hamarneh. Vulnerability analysis of chest x-ray image classification against adversarial attacks, 2018. arXiv: 1807.02905 [cs.CV] (cited on pages 13, 24).

[TEC02] Lyn C. Thomas, David B. Edelman, and Jonathan N. Crook. *Credit Scoring and Its Applications*. en. Society for Industrial and Applied Mathematics, January 2002. ISBN: 978-0-89871-483-8 978-0-89871-831-7. DOI: 10.1137/1.9780898718317. URL: http://epubs.siam.org/doi/book/10.1137/1.9780898718317 (visited on 01/17/2020) (cited on pages 10, 23, 29).

[TPG+17] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017 (cited on pages 7, 54).

[VC21] Gaël Varoquaux and Veronika Cheplygina. How i failed machine learning in medical imaging–shortcomings and recommendations. *arXiv preprint arXiv:2103.10292*, 2021 (cited on page 84).

[VGD+19] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. *arXiv preprint arXiv:1904.02920*, 2019 (cited on page 59).

[VGV+21] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021 (cited on pages 21, 54, 58, 64, 73, 75, 87).

[VMU+20a] M Vollmer, S Mishra, H Unwin, A Gandy, T Melan, V Bradley, H Zhu, H Coupland, I Hawryluk, M Hutchinson, et al. Report 20: a sub-national analysis of the rate of transmission of covid-19 in italy, 2020 (cited on page 26).

[VMU+20b] Michaela A C Vollmer, Swapnil Mishra, H Juliette T Unwin, Axel Gandy, et al. Report 20 : Using mobility to estimate the transmission intensity of COVID-19 in Italy : A subnational analysis with future scenarios. Technical report May, Imperial College COVID-19 Response Team, 2020, page 35 (cited on pages 14, 106).

[WG21] Guile Wu and Shaogang Gong. Collaborative optimization and aggregation for decentralized domain generalization and adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6484–6493, October 2021 (cited on page 21).

[WLL+21] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. Generalizing to unseen domains: a survey on domain generalization. *arXiv preprint arXiv:2103.03097*, 2021 (cited on page 21).

[Wor20] World Health Organisation. *a Coordinated Global Research Roadmap: 2019 Novel Coronavirus*, number March. 2020. ISBN: 9789241549837 (cited on pages 13, 105).

[WPL+17a] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. Chestx-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. DOI: 10.1109/cvpr.2017.369. URL: http://dx.doi.org/10.1109/CVPR.2017.369 (cited on pages 24, 87).

[WPL+17b] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. Chestx-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. DOI: 10.1109/cvpr.2017.369. URL: http://dx.doi.org/10.1109/CVPR.2017.369 (cited on page 72).

[WRK20] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020 (cited on pages 20, 162).

[WTF+20] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: investigating and improving multi-task optimization in massively multilingual models. *arXiv preprint arXiv:2010.05874*, 2020 (cited on pages 73, 162).

[WTW19] Aria Wang, Michael Tarr, and Leila Wehbe. Neural taskonomy: inferring the similarity of task-derived representations from brain activity. *Advances in Neural Information Processing Systems*, 32, 2019 (cited on page 22).

[XM12] Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012 (cited on page 6).

[XY11] Qian Xu and Qiang Yang. A survey of transfer and multitask learning in bioinformatics. *Journal of Computing Science and Engineering*, 5(3):257–268, 2011 (cited on page 54).

[YCW+20] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: a diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020 (cited on page 54).

[YKG+20] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020 (cited on pages 73, 75, 162).

[YPC+19] Li Yao, Jordan Prosky, Ben Covington, and Kevin Lyman. A strong baseline for domain adaptation and generalization in medical imaging, 2019. arXiv: `1904.01638 [cs.CV]` (cited on pages 12, 84).

[YRZ+20] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Advances in neural information processing systems*, 33:8588–8601, 2020 (cited on page 6).

[YZW+20] Zifeng Yang, Zhiqi Zeng, Ke Wang, Sook-San Wong, Wenhua Liang, Mark Zanin, Peng Liu, Xudong Cao, Zhongqiang Gao, Zhitong Mai, et al. Modified seir and ai prediction of the epidemics trend of covid-19 in china under public health interventions. *Journal of Thoracic Disease*, 2020 (cited on page 26).

[Za21] Haoran Zhang and al. An empirical framework for domain generalization in clinical settings. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 279–290, 2021 (cited on page 25).

[ZH19] Shayan Zamani and Hadi Hemmati. Revisiting hyper-parameter tuning for search-based test data generation. In Shiva Nejati and Gregory Gay, editors, *Search-Based Software Engineering - 11th International Symposium, SSBSE 2019, Tallinn, Estonia, August 31 - September 1, 2019, Proceedings*, volume 11664 of *Lecture Notes in*

Computer Science, pages 137–152. Springer, 2019. DOI: 10.1007/978-3-030-27455-9_10. URL: https://doi.org/10.1007/978-3-030-27455-9%5C_10 (cited on page 44).

[ZK16] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016 (cited on pages 76, 156).

[ZSS+18] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018 (cited on pages 59, 64, 67, 73, 89).

[ZXH+20] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International conference on machine learning*, pages 11278–11287. PMLR, 2020 (cited on page 20).

[ZY17] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017 (cited on page 54).

[ZYH+20] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation, 2020. arXiv: 2003.06054 [cs.CV] (cited on page 21).

[ZYJ+19] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019 (cited on pages 20, 162).