PhD-FSTM-2026-008
The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 09/01/2026 in Esch-sur-Alzette

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Lena Maria HARTMANN
Born on 20 March 1997 in Berlin (Germany)

MULTI-OBJECTIVE LEARNING IN FEDERATION

Dissertation defence committee

Dr Grégoire DANOY, dissertation supervisor
Assistant Professor, Université du Luxembourg

Dr Frédéric GUINAND
Professor, Université Le Havre Normandie

Dr Pascal BOUVRY, Chairman
Professor, Université du Luxembourg
Dean of the Faculty of Science, Technology and Medicine

Dr Ann NOWÉ
Professor, Vrije Universiteit Brussel

Dr Bernabé DORRONSORO
Professor, Universidad de Cádiz

Abstract

Federated Learning (FL) is a powerful distributed computing paradigm, developed for scenarios where data originates in distribution and cannot be shared due to privacy, confidentiality, or hardware constraints. Training a machine learning model in such circumstances was previously limited to training a separate model on each data silo, without any knowledge of the data available elsewhere. This approach generally leads to models of lower quality than centralised baselines, trained on a full centralised dataset. By enabling collaborative learning without sharing raw data, Federated Learning promises to unlock the more general insights found in larger datasets for the distributed setting. Indeed, this strategy has already been widely adopted in the industry, including for deployment on mobile phones and in the finance and health sectors, typically to overcome privacy and confidentiality constraints. However, theoretical challenges remain, often connected to the failure of existing federated algorithms to account for the true complexity of real-world problems. As the capability of machine learning algorithms and hardware grows, so too does the scope for distributed use cases, requiring the adaptation of the federated paradigm to such emerging challenges. Multi-objective modelling is a well-recognised approach to modelling the complexity of the real world with its frequently conflicting requirements. Despite its broad applicability, this direction of research has received very little attention to date.

This thesis explores the opportunities and challenges of integrating multi-objective methods with Federated Learning, with a focus on facilitating multi-objective learning in federation. In a first contribution, we provide a comprehensive survey of the literature combining multi-objective and Federated Learning techniques and propose a first systematic taxonomy of the field. We categorise existing works into this taxonomy and identify open areas of research, noting that federated multi-objective learning (FMOL) in particular remains underexplored.

Following this insight, we propose a first novel framework and an algorithm, respectively, for two distinct FMOL settings. In the first setting, previously unaddressed in the literature, distributed parties collaborate under the control of a server to find a full spectrum of trade-off solutions. Our proposed framework, MOFL/D, formalises a general approach based on decomposition, a well-established strategy from the field of multi-objective optimisation. In the second setting, participants assign different pre-defined importance preferences to the objectives of the problem. Each party is interested in finding a single solution that matches its own preferences, leading to the challenge of aligning models with conflicting preferences. We propose an algorithm, FedPref, that finds a personalised model for each participant, modulating collaboration during the learning process based on similarity.

Next, we consider how to validate FMOL algorithms appropriately. We argue that the currently predominant benchmarking problems fail to represent the true difficulty of multi-objective learning, lacking inherent conflict between objectives. Consequently, we propose a new class of accessible, flexible, and scalable benchmarking problems, derived from the field of fair machine learning (Fair ML), that are known to contain such conflicts. We demonstrate the instantiation of a range of these Fair ML benchmarks and show their use on state-of-the-art algorithms.

In a final chapter, we look towards the future, examining the potential of Federated Learning as a tool in the space domain. We identify a challenging potential use case, envisioning the use of FL algorithms to establish ad-hoc collaboration between satellites performing orbital edge computing. We discuss the state of the art in the field of Federated Learning as relating to this use case, and point out gaps where further research is required. Aside from algorithmic challenges, one crucial gap is a lack of existing standards to facilitate the exchange of machine learning models in spacecraft communications. We consider a potential pathway towards the rapid establishment of such standards.

Key words: Federated Learning, multi-objective learning

Acknowledgements

First and foremost, I would like to thank my supervisor, Grégoire Danoy, for guiding me with kindness, patience and good humour to become a better academic and a better person. This work would have been far less successful and less enjoyable without his unwavering support and encouragement.

I am grateful to Pascal Bouvry and the ILNAS for giving me the opportunity to complete my PhD as a member of this project, and I would like to thank everyone involved in it. On the ILNAS side, I am particularly grateful to Lucas Cicero and Natalia Vinogradova for their collaboration on all things standardisation, and to Jean-Philippe Humbert for his constant encouragement. Special thanks must also go to my fellow PhD students on the project, Manuel and Hedieh - I could not have wished for better officemates and friends, throughout the many late nights in and out of the lab.

Beyond my project, I would like to thank all members of the PCOG team, past and present, that I've had the pleasure to meet, work, and run with. Special thanks to Gabriel, Florian, Gwen, and Yi-Nung for their example, as well as companionship, cultural exchange, and involvement in the occasional mischief.

Finally, I owe thanks to my family and friends for their encouragement and support of my choices, even when they take me further from them. I am grateful for the forbearance and moral support or my parents, Petra and Jens, and for the companionship in this life of my favourite twin, Anne.

To everyone mentioned here, and the countless other teachers, mentors, colleagues, athletes, family, and friends who have shaped me in my journey here, thank you.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

CCSDS . . . . . Consultative Committee for Space Data Systems

DDP . . . . . Difference of Demographic Parity
DMC . . . . . Deterministic Minecart
DP . . . . . Demographic Parity
DST . . . . . Deep-Sea Treasure

EO . . . . . Equality of Opportunity
EOD . . . . . Equalised Odds
ESA . . . . . European Space Agency

FL . . . . . Federated Learning
FMOL . . . . . Federated Multi-Objective Learning
FT . . . . . Fine-Tuning

IGD . . . . . Inverted Generational Distance

LEO . . . . . Low Earth Orbit

MEO . . . . . Medium Earth Orbit
MGDA . . . . . Multi-objective Gradient Descent Algorithm
MO-HC . . . . . Multi-Objective HalfCheetah
MO-HPO . . . . . Multi-Objective HyperParameter Optimisation
MO-LLc . . . . . Multi-Objective Lunar Lander - continuous
MO-NAS . . . . . Multi-Objective Neural Architecture Search
MOEA . . . . . Multi-Objective Evolutionary Algorithm
MOFL . . . . . Multi-Objective Federated Learning
MOFL/D . . . . . Multi-Objective Federated Learning with Decomposition
MOLL . . . . . Multi-Objective Lunar Lander
MOLP . . . . . Multi-Objective Learning Problem
MOML . . . . . Multi-Objective Machine Learning

MOO . . . . . Multi-Objective Optimisation
MOO/D . . . . . Multi-Objective Optimisation with Decomposition
MORL . . . . . Multi-Objective Reinforcement Learning
MTL . . . . . Multi-Task Learning

NAS . . . . . Neural Architecture Search
NNEF . . . . . Neural Network Exchange Format
NSGA-II . . . . . Non-dominated Sorting Genetic Algorithm II

OEC . . . . . Orbital Edge Computing
ONNX . . . . . Open Neural Network eXchange format

PF . . . . . Pareto Front
PFL . . . . . Personalised Federated Learning

RL . . . . . Reinforcement Learning

VFL . . . . . Vertical Federated Learning

1 | Introduction

Contents

1.1 | Motivation

Machine learning methods can find and encode patterns in the world that are difficult to recognise or describe for human beings. In recent years, research in the field has progressed along with improved hardware capabilities and increased compute resources. This synergy has led to a series of impressive leaps forward – most prominently in the development of large language models, – in turn driving a rapid acceleration in the deployment of AI tools across a growing range of use cases and domains.
However, standard ML methods have begun to hit conceptual barriers in certain use cases: most ML algorithms require large amounts of data to thrive and benefit from inputs that represent the full diversity of the problem space. However, in many real-world use cases potential training data is owned by different parties that are unwilling or unable to share their data. Such restrictions may arise in many different domains, for example:

- When training a model to aid medical personnel in the evaluation of diagnostic imaging data, multiple hospitals may wish to collaborate to build a more representative sample base. Patient data is protected by privacy legislation in many locales, so hospitals are typically unable to share such data with third parties.
- Mobile phone providers may wish to train models to enhance aspects of the user experience, e.g. by continuously improving predictive keyboard functionalities. Individual user data contains sensitive personal information, so should not be collected systematically outside of the user's personal device.
- Multiple financial stakeholders may wish to collaborate to build a model that e.g. predicts the risk of potential customers defaulting on a credit loan. Customer data is confidential and should not be share across different entities.
- Multiple telecommunications providers may wish to train a lightweight model to better allocate satellite resources based on expected demand. In this case, resource limitations form the barrier to sharing training data, as the limited power available to each satellite cannot usually support extensive additional satellite-to-ground or satellite-to-satellite transmissions.

These examples illustrate how a range of different constraints, including privacy, confidentiality, and technological restrictions, may limit how a machine learning model can be trained using conventional centralised methods. The Federated Learning paradigm offers a way of overcoming these challenges, by facilitating training in such settings in a decentralised manner that requires no sharing of raw data. Instead, separate models are trained locally on each available dataset, and only the resulting model parameters are exchanged. Though originally designed to mitigate privacy concerns, the method has also shown great success in other use cases, including communication-restricted settings such as drone networks [Bri20] or computationally costly settings such as the tuning of large language models [Che23].

However, as Federated Learning is adopted in increasingly diverse applications and real-world use cases, new challenges are emerging, many linked to the need to balance different conflicting requirements: (i) Heterogeneity between participants caused by data imbalances or differing hardware capabilities can lead to divergent local models that cannot easily be aggregated without loss of model utility [Kar19]. Designing mitigation strategies for this raises the problem of fairness – the choice between sacrificing the performance of some individual clients or that of the global model. (ii) The cost of FL in terms of communication and computation resources scales with the size of the model and the number of update messages; yet reducing either may come at the cost of decreasing model utility [Zhu21a]. (iii) Strategies for mitigating privacy leakage, the problem of exposing confidential information to potential attackers through client updates, may degrade other aspects of the federated system in turn. For example, adding noise to client updates may obscure sensitive information effectively, but reduce model performance as well [Gen24].

All these scenarios can be modelled as multi-objective problems, with each problem-specific performance metric represented as a separate objective. Under this multi-objective perspective, problems are solved with explicit consideration for several characteristics, potentially conflicting, and solutions can represent different optimal trade-offs between all objectives. As such, the approach can assist users in making informed decisions about complex FL problems by presenting explicit choices where a single-objective approach would yield none. Indeed, these general advantages of multi-objective methods have been recognised across disciplines, and the field of multi-objective optimisation (MOO) has been thriving for decades [Coe25]. This success opens another interesting avenue of research in connection with federated learning: deploying FL methods to facilitate multi-objective learning in distribution, where problems would otherwise be difficult to solve for participants that cannot share local training data. This has direct applications to many naturally federated settings: in medical use cases, multi-objective modelling could permit balancing the likelihood of correct disease identification with the risk of false positives. When training personalised behaviour on mobile devices, it could give explicit control over the trade-off between privacy and model performance, and financial prediction tools could be trained to balance potential risk and reward. Finally, communication satellites predicting expected traffic might be able to trade off predictive accuracy and resource risk.

Clearly, federated multi-objective methods could make a powerful impact in practice, giving increased control over and understanding of frequently occurring trade-offs between different objectives to users, as centralised multi-objective methods have done before.

1.2 | Research questions

Though research in the field of Federated Learning is advancing rapidly, and multi-objective optimisation contains a well-established body of literature, the overlap of both fields has been explored little to date. Nevertheless, the potential of multi-objective techniques to enhance FL methods has been recognised in principle, and demonstrated in several instances [Hu22a; Meh22; Zhu22]. The multi-objective modelling approach also retains its well-recognised advantages when applied to general problems in the distributed setting. However, this approach raises unique challenges when combined with Federated Learning [Har25b]. These observations lead to our first research question, concerned with systematically exploring the patterns and challenges in combining multi-objective and federated learning techniques.

The second research question targets the gap in federated learning methods that can be used to solve arbitrary multi-objective learning problems. As the following chapters will show, our study of the first research question reveals a number of distinct scenarios for federating multi-objective problems, arising from different assumptions about the roles of clients and server. Several of these scenarios have clear corresponding use cases in the real world, but have never been tackled before in the literature.

The third research question concerns the validation of federated algorithms designed to solve multi-objective problems. Representative benchmarks are crucial to accurately evaluate the performance of proposed algorithms and place them in the context of previous work in the field. However, the question of benchmark design for federated multi-objective learning algorithms has not formally been addressed in the literature. Existing work relies on a single class of problems, transferred from related research on centralised algorithms, for benchmarking.

In the final research question of this thesis, we take a broader view of the potential trajectory of our research. Initial application of the Federated Learning paradigm was largely focused on privacy-related use cases. With recent progress in the development of smaller and more powerful hardware components, a new application front for FL is opening up, focused on distributed devices where data sharing is restricted by resource constraints. One promising use case of this type comes from the aerospace domain. Recent trends in the design of space missions are moving towards small multi-satellite mission configurations, with the long-term goal of developing full autonomy. Federated Learning could provide a resource-efficient solution for on-board machine learning, enabling continuous improvement for satellites based on real-world data. Our research is focused on examining this use case, identifying relevant gaps in the current state of the art and discussing potential solution approaches, including with respect to standardisation efforts.

- ➢ **Q1** What are the commonalities, differences and challenges in combining multi-objective methods with federated learning?
- ➢ **Q2** How can multi-objective learning problems be solved in federation?
- ➢ **Q3** How can federated multi-objective learning algorithms be validated in a general way?
- ➢ **Q4** What other challenges currently hinder the application of FL methods in complex real-world use cases, such as the space domain?

1.3 | Thesis structure and contributions

This thesis is structured as follows: In Chapter 2, we address Question 1 by exploring the existing work in the literature that integrates Federated Learning and multi-objective techniques. We offer a first clear and systematic overview of the different ways the two fields can be integrated. We propose a first taxonomy on the use of multi-objective methods in connection with Federated Learning, providing a targeted survey of the state-of-the-art and proposing unambiguous labels to categorise contributions. Given the developing nature of this field, our taxonomy is designed to provide a solid basis for further research, capturing existing works while identifying gaps and anticipating future additions. Finally, we outline open challenges and possible directions for further research.

Question 2 is addressed in Chapters 3 and 4. Chapter 3 considers a scenario – not previously addressed in the literature – where fully cooperative clients wish to solve a multi-objective problem under the guidance and control of a server. We propose a first general framework for Federated Multi-objective Learning, based on decomposition, to compute a Pareto front of solutions across a federated system. This framework addresses the *a posteriori* type of multi-objective problem, where user preferences are not known during the optimisation process. We present an instantiation of the framework and validate it through experiments on a set of multi-objective benchmarking problems that are extended from well-known single-objective benchmarks.
In Chapter 4, we tackle a different scenario, assuming now that each client holds an individual set of personal preferences over the objectives that is unknown to the server. We identify the novel challenge of *preference heterogeneity* arising from such a setting, and propose a first Personalised Federated Learning algorithm to solve it. This algorithm, known

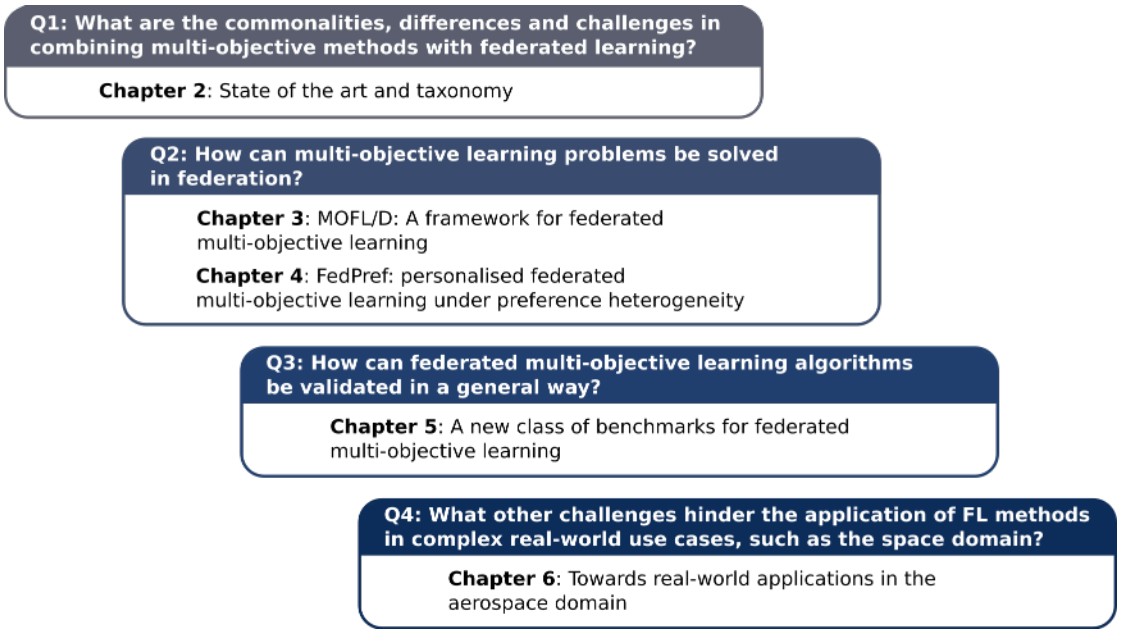


Figure 1.1: The structure of the main body of this thesis.

as FedPref, is based on similarity-based clustering and weighted aggregation. We validate FedPref on a comprehensive set of experiments across different preference distributions, showing that it is robust and effective in a variety of use cases. Finally, we introduce the use of multi-objective metrics to evaluate the performance of a FL algorithm under preference heterogeneity at system level.

Chapter 5 continues on the theme of evaluation by addressing Question 3. The evaluation of novel methods requires suitable benchmarks that are representative of the problem setting. In Federated Learning, benchmarks are commonly transferred from centralised settings without modification. In this chapter, we show that this practice is not sufficient for FMOL: in one natural setting, where federated clients have heterogeneous preferences over multiple objectives, the most commonly used class of benchmarks can be solved easily even by baseline algorithms, in apparent contrast to the difficulty of the problem in the non-federated setting. Following this insight, we introduce a different, more challenging class of benchmarking problems, derived from the field of fair machine learning (Fair ML). These benchmarks are adaptable, easy to implement, permit diverse model architectures and different (numbers of) objectives, include a range of different well-established datasets, and do not require special adaptation of the federated algorithm. We run state-of-the-art algorithms on several instances of our proposed benchmarks, showing their versatility and applicability to a range of common Federated Multi-Objective Learning scenarios.

Finally, we address Question 4 in Chapter 6, exploring the potential application of Federated Learning algorithms to multi-satellite missions consisting of small, resource-limited spacecraft. Federated Learning is a promising distributed computing approach in this context, allowing multiple satellites to collaborate efficiently in training on-board machine learning models. Though recent works on the use of Federated Learning in orbital edge computing have focused largely on homogeneous satellite constellations, Federated Learning could also be employed to allow heterogeneous satellites to form ad-hoc collaborations, e.g. in the case of communications satellites operated by different providers. Such an application presents additional challenges to the Federated Learning paradigm, arising largely from the heterogeneity of such a system. We offer a systematic review of these challenges in the context of the cross-provider use case, giving a brief overview of the state-of-the-art for each, and providing an entry point for deeper exploration of each issue. In addition, we discuss how standardisation could pave the way for the deployment of such novel approaches. In particular, we examine the challenges of communicating such models from ground to space and between spacecraft in a standardized, unambiguous way. We consider how existing communication protocols and transfer formats could be employed to achieve this, and suggest where modifications may be necessary.

2 | State of the art and taxonomy

Contents

Recent works in the literature have begun to combine federated learning (FL) with multi-objective (MOO) methods to address a wide range of challenges. However, the broader context of the intersection between MOO and FL has not yet been discussed. This chapter aims to provide a first such systematic overview, identifying general challenges and parallels, and formulating a novel taxonomy to classify existing work while highlighting open directions of research.

Many FL strategies already use (linear) combinations of multiple functions as objectives, but do not consider the problem from a multi-objective angle. The first works to explicitly introduce multi-objective methods to Federated Learning aimed to improve federated aggregation and introduce fairness between clients [Hu22b], followed by approaches introducing other, system-wide aggregation parameters [Meh22]. Another early adoption of MOO was in hyperparameter optimisation for FL [Zhu20a]. More recently, research has also begun into supporting the inverse scenario: developing strategies to federate the solving of multi-objective problems by distributed clients, e.g. [Yan23b][Har23a]. The contributions of this chapter can be summarised as follows:

- We propose a novel taxonomy of algorithms combining MOO methods and FL, offering a unified terminology for works at the intersection of two previously largely separate fields with separate naming conventions.
- We present a thorough review of the state of the art, categorising and contrasting existing works.
- We highlight open questions and offer perspectives on open avenues for future research.

The rest of this chapter is organised as follows: Section 2.1 reviews important notions from the fields of FL and MOO. Section 2.2 introduces our taxonomy, discussing in detail each category and relevant works from the literature.

2.1 | Background

In this section, we briefly introduce fundamental concepts from the fields of federated learning and multi-objective optimisation to provide the necessary background for the remainder of the survey.

2.1.1 | Federated Learning

The Federated Learning [McM17b] paradigm was originally designed to solve arbitrary (neural network-based) machine learning problems in a difficult distributed setting. This setting is characterised by (i) the available data originating in distribution, with no control over the composition of the resulting datasets, and (ii) a restriction on transmitting private client information, including raw training data, between participants. FL overcomes the constraint introduced by (ii) by training separate local models in distribution on each dataset holder, or client, as originating from (i), and aggregating only the resulting models across clients – see Figure 2.1.

A more detailed general framework of the Federated Learning strategy is presented in Alg. 1, with colours highlighting the correspondence of code segments to different levels of the federated system (to be presented in detail in Section 2.1.3). First, the federated system is initialised with the identity of the server, a list of participating clients, and the definition of the underlying learning problem to be solved. Additional hyperparameters are passed depending on the specific algorithm, defining e.g. the architecture of the neural network to be trained, a client sampling rate, gradient thresholds, or any other parameter required by the algorithm. Then, the local learning process begins. During each federated training round, a set of clients is selected for participation. These clients each carry out local training and return the resulting models to the server. These local models are aggregated periodically by the server into a single global model incorporating the locally learned information. The global model is then passed back to the local clients to continue the next local training round. Expressed formally, the FL process aims to find a global model θ that generalises to all available data, i.e.

$$minimise_\theta f(\theta, \mathcal{D}), \tag{2.1}$$

where $\mathcal{D} := \bigcup_i^n \mathcal{D}_i$, with $\mathcal{D}_i$ the dataset of the i-th client. Imbalances between client datasets, as can be caused by characteristic (i), represent a significant challenge to the

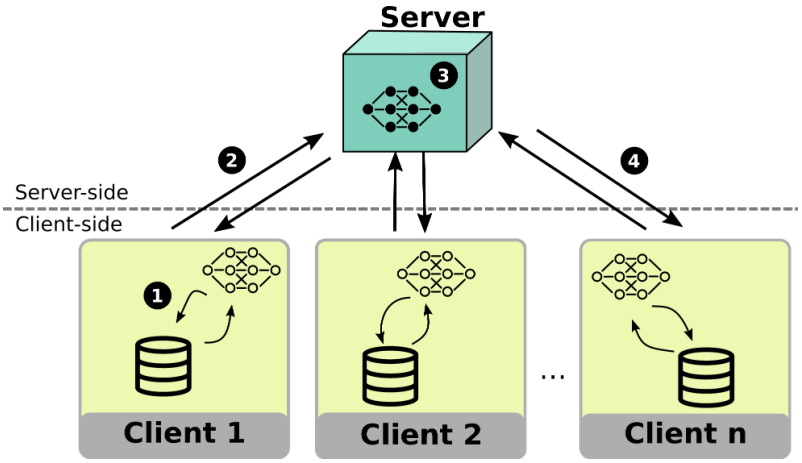


Figure 2.1: The FL paradigm. During each round, clients perform local model training (1), then transmit their local models to the server (2) for aggregation into a single global model (3). The global model is returned to the clients (4) to begin the next training round.

model aggregation step of FL algorithms. Indeed, any type of heterogeneity between clients, e.g. in terms of hardware capability or feature distribution, may have an adverse impact on the convergence of the federated model. Mitigating the impact of various types of client heterogeneity remains an active field of study. Other major research topics in FL include the reduction of resource consumption – mainly computing and communication cost – and how to protect against malicious actors. For a comprehensive overview of the state of the art in the field, we refer to [Kai21a].

Algorithm 1 The general Federated Learning framework.

Input: Server, list of clients, local learning problem.
Parameter: Optional list of hyperparameters.
Output: Global model θ.

```
1: Initialise system parameters.
2: while stopping condition not satisfied do
3:     for all participating clients do
4:         while local stopping condition not satisfied do
5:             Perform training on local data.
6:         end while
7:         Transmit local model to server.
8:     end for
9:     Aggregate local models to obtain new global model.
10:    Return global model to clients.
11: end while
12: return global model.
```

2.1.2 | Multi-objective optimisation

Multi-objective optimisation is concerned with solving problems in the presence of more than one objective. As an example, consider the problem of selecting hyperparameters for a neural network to simultaneously maximise model utility and minimise the cost of training. Instead of a single objective $f(x)$, such a multi-objective problem is expressed as a vector of n objectives $\vec{f}(x) := (f_1(x), \ldots, f_n(x))^T$. Note that individual objectives can conflict, i.e. in general no single solution can optimise all objectives simultaneously. Instead, MOO methods typically focus on identifying solutions that represent an optimal trade-off between objectives, where objective values are balanced so that no single objective can be improved without sacrificing the performance of another. Such trade-off solutions are known as *Pareto-optimal*. Pareto optimality can be difficult to determine in practice, where the optimal values achievable for each objective are unknown, so the weaker notion of *Pareto-dominance* is commonly used instead. A solution x is said to Pareto-dominate another solution y iff it outperforms y in at least one objective while matching or improving the value of all others. Formally,

$$x \succ_P y \iff \exists j f_j(x) > f_j(y) \wedge \forall i f_i(x) \geq f_i(y) \tag{2.2}$$

for a maximisation problem. Pareto-optimal solutions are not dominated by any others. The set of such solutions is known as the *Pareto front* (see Fig. 2.2). Most MOO algorithms are either designed to find such a Pareto front, or a single solution based on predefined requirements such as user preferences. A wide range of algorithmic approaches exists for both variants, tailored to different problem characteristics. In this chapter, we will discuss relevant MOO strategies as they appear; for a comprehensive overview we refer to [Tal09].

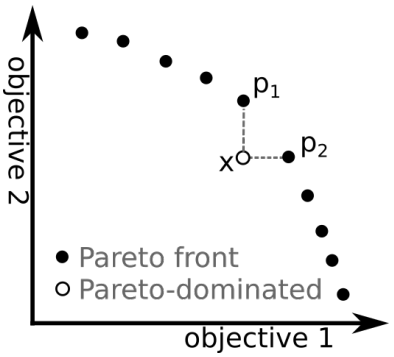


Figure 2.2: Pareto front and Pareto dominance. Shaded markers represent solutions on the Pareto front of a bi-objective maximisation problem; x is Pareto-dominated by p_1 and p_2.

2.1.3 | Integrating multi-objective methods and Federated Learning

We note that multi-objective methods can be integrated with FL at different levels of the federated system, each with distinct implications for the algorithmic components involved. Based on this insight, we propose a three-level view of the federated system – see Fig. 2.3 and corresponding colours in Alg. 1. Adding multi-objective methods on

top of a federated algorithm necessitates no modification of the underlying federation or local learning process; an example for such a method is offline hyperparameter tuning with respect to multiple requirements. On the other hand, introducing multi-objectivity at the federated level, e.g. for model aggregation on the server, forces adaptation at the top level as well: any hyperparameter algorithm running on the federated system must accommodate new parameters introduced by multi-objective methods. Finally, adding a multi-objective perspective to the lowest level in Fig. 2.3 – the client level – requires modifications across the entire system: (i) The local learning algorithm on each client must handle multi-objective problems; (ii) the federated algorithm must aggregate client submissions, which may include multi-objective gradients or be influenced by heterogeneous client objectives, and (iii) any hyperparameter must be adjusted once again.

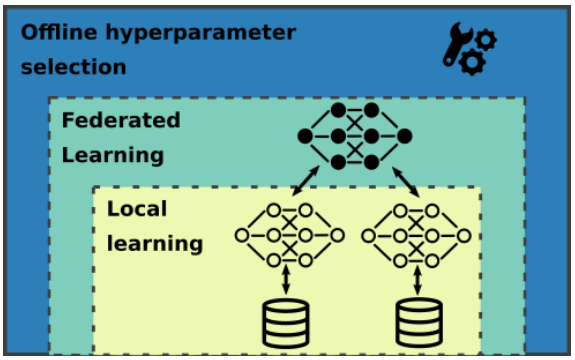


Figure 2.3: Relation of major categories of the taxonomy. Multi-objective methods can be integrated at different levels of the federated system: in the local learning process of clients, at system-level in the federated algorithm, or outside of the federated system.

2.2 | Taxonomy: multi-objective methods in FL

In this section, we introduce our proposed taxonomy, discussing each category and the related existing work. The full taxonomy is shown in Fig. 2.4. A first fundamental distinction is the purpose that multi-objective and federated methods each serve in an algorithm. We can identify two main broad categories: one where MOO methods are applied to enhance the functionality of a federated system, and the inverse, where FL is used in support of solving a general multi-objective problem in distribution. We refer to these categories as *Multi-Objective Federated Learning (MOFL)* and *Federated Multi-Objective Learning (FMOL)*, respectively, to indicate the different chaining of strategies. MOFL covers the majority of existing research, and is notably precisely equivalent to the top two layers as shown in Fig. 2.3 and introduced in Section 2.1.3. Works in this section can accordingly be divided further into top-level and federation-level methods, and will be discussed as such in the following sections. FMOL, in contrast, corresponds to the lowest layer in Fig. 2.3, and extends the "standard" FL scenario, where Federated Learning is used to solve an arbitrary learning problem in distribution, to include multi-objective learning problems.

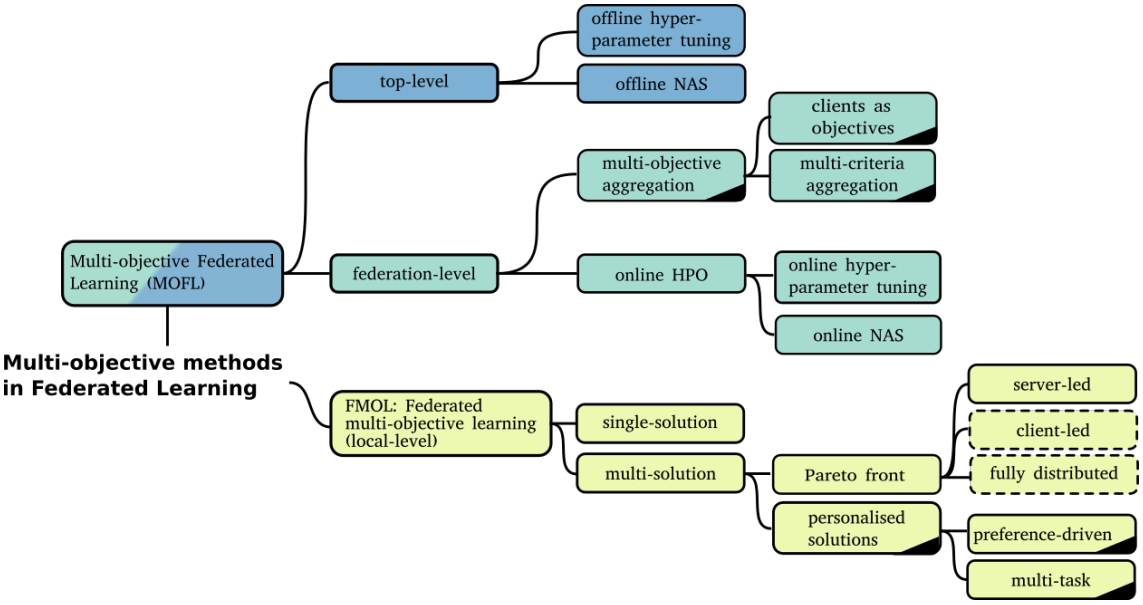


Figure 2.4: Proposed taxonomy. Colours denote the level of the federated system where MO methods are integrated (see Fig. 2.3 and Sec. 2.1.3). Some categories arise from the unique properties of the FL setting; these are marked by a shaded corner. Categories in dashed boxes are currently unexplored in the literature.

2.2.1 | Multi-objective federated learning at top level

Methods at the top level of a federated system, as defined in Fig. 2.3, are decoupled from the federated learning and aggregation process and can treat the federated algorithm as a black-box system. As such, this class of algorithms is arguably the least specific to the FL context, since modifications at this level require no particular adaptation to the federated setting. Current work can largely be divided into two major applications: multi-objective neural architecture search (MO-NAS), focused on optimising the architecture of a neural network with respect to multiple objectives, and more general multi-objective hyperparameter tuning, where other hyperparameters of the federated system are tuned. Both types typically employ population-based multi-objective strategies, known to offer effective search space exploration.

2.2.1.1 | Offline hyperparameter tuning

Multi-objective hyperparameter tuning can find algorithm parameters for additional requirements beyond the utility of the global model. Depending on the use case, FL systems may face challenges such as privacy restrictions, resource limitations, or malicious attacks. This approach allows users to explicitly model such requirements and make informed choices about the trade-offs inherent to different solutions.

[Kan24b] assert that optimising hyperparameters solely for model performance may expose the federation to a risk of data leakage. The proposed mitigation approach optimises the three objectives of model performance, training cost and privacy leakage simultaneously. This algorithm, derived from NSGA-II [Deb02], a well-known population-based metaheuris-

tic, is designed to find a Pareto front of possible configurations representing different trade-offs between these objectives. [Mor24] also introduce a second objective in addition to the model accuracy, based on the mean amount of data transmitted and received by clients. This approach is designed to optimise a large number of hyperparameters and algorithmic choices, including the number of local training steps, the number of bits used to encode local updates, and whether clients submit gradient or weight updates. All variables are optimised using a hybrid of NSGA-II and an estimated distribution-based algorithm (EDA). [Gen24] formulate a similar strategy, also using NSGA-II, but considering the four objectives of minimising global model error rate, the variance of model accuracy, the communication cost, and a privacy budget.

2.2.1.2 | Offline neural architecture search

Neural architecture search aims to optimise the structure of a neural network for given objectives. Federated NAS can be seen as an inherently multi-objective problem [Zhu21a], as changes to the model structure impact not only the model utility, but also other aspects of the federated system, such as the communication and training cost. One of the first works on multi-objective federated neural architecture search [Zhu20a] proposes an offline federated NAS algorithm that constructs models with the two objectives of minimising the validation error obtained by the model, and the cost of communicating the model. Solutions are once again generated using NSGA-II. The same problem is tackled in [Cha22], but with the use of another type of multi-objective evolutionary algorithm (MOEA) instead of NSGA-II to improve the exploration of the multi-objective search space.
Federated split learning is a related problem, where partial blocks of the global model are assigned to clients, with blocks of different size assigned to clients depending on the available resources. [Yin23] propose to optimise this splitting decision, along with communication bandwidth and computing resource allocation, as a multi-objective problem, minimising training time and energy consumption of the system. The proposed algorithm yields a Pareto front of solutions using a hybrid of NSGA-III and a generative adversarial network trained to identify configurations generating Pareto-dominated solutions. Research on offline MO-NAS algorithms for FL is arguably more advanced than other areas of MOFL, as existing approaches can be applied to the federated setting without change. The main challenge remains the high computational cost of these methods.

2.2.2 | Multi-objective federated learning at federation-level

MOO methods can also be integrated with FL at the server-level to solve challenges inherent to the FL paradigm – a brief overview of representative works from the literature is presented in Table 2.1. The majority of existing works focus on one of two design aspects of a federated system: the aggregation strategy used on the federated server, and the selection of relevant hyperparameters for the FL algorithm. We discuss both separately, beginning with multi-objective aggregation.

2.2.2.1 | Multi-objective aggregation

The aggregation of local model updates by the server can be modelled as a MOO problem, permitting the use of more than one criterion for computing the global model. This

Table 2.1: Comparison of selected MOFL algorithms. Each row lists the level of the federated system where multi-objective notions are introduced, as well as the method used to solve the multi-objective problem.

Reference	Taxonomy label	System level	MOO method	Objectives
[Hu22b]	Clients as objectives	federation-level	MGDA	Local model utilities
[Ju24]	Clients as objectives	federation-level	dynamic preferences	Fairness, convergence
[Meh22]	Multi-criteria aggregation	federation-level	obj.-contribution scoring	Arbitrary system objs.
[Zhu22]	online MO-NAS	federation-level	NSGA-II	Global model utility, evaluation speed
[Kan24b]	offline MO-HPO	top-level	NSGA-II	Model utility, training cost, privacy leakage

multi-objective version of federated aggregation can be formulated in general terms as follows:

$$min_{\theta}(f_1(\theta), \dots, f_n(\theta)),^T \tag{2.3}$$

where θ is the global model and f_i is the loss function of the i-th objective. Solving this problem typically translates to finding optimal aggregation weights λ_i to compute the global model from the local models:

$$min_{\lambda_1 \dots, \lambda_n}(f_1(\theta), \dots, f_n(\theta)),^T, \text{ with } \theta = \sum_{i}^{n} \lambda_i \theta_i. \tag{2.4}$$

The literature on FL algorithms with multi-objective aggregation can be categorised based on the nature of the objectives [Kan24a]. One line of work derives objectives from the performance of individual clients; the other uses objectives that describe the federation as a whole. This distinction is significant, as the different mathematical properties of these variants permit the use of different multi-objective methods. The following sections discuss both types in detail.

Clients as objectives. These algorithms consider the performance of individual clients and the global model as separate objectives. In client-heterogeneous settings, this approach can balance the interests of both the clients and the general system. This perspective enables explicit fairness guarantees for selfish participants, ensuring that the performance of individual clients is not sacrificed for that of the system in computing the global model. Crucially, performance criteria in this class of MOFL problems are tied directly to the client models and thus differentiable with respect to model parameters. As such, they can be solved efficiently using gradient-based multi-objective algorithms such as the classical multi-gradient descent algorithm (MGDA) [Dés12], established in the field of MOO.
The FedMGDA+ algorithm [Hu22b] leverages this insight, defining the performance of each participating client as a separate objective. Using MGDA yields aggregation weights

for a gradient representing a common direction of descent for all clients, thus guaranteeing that no client suffers a reduced performance by participating in an aggregation step. An added constraint on the divergence of aggregation weights serves as protection against false updates by malicious participants. The FedMC+ algorithm [She25] is also designed to reconcile individual client updates and the global model in the presence of heterogeneous data. A secondary objective, minimising conflict between the global and local gradients, is introduced during the aggregation step and solved by transformation into a convex optimisation problem. [Cui21] formulate the aggregation step as a parameterised min-max optimisation problem. Fairness constraints serve to optimise model utility for the single worst-performing client while ensuring that (i) the utility of all clients improves, and (ii) no client improves much less than another. The solution obtained from this formulation is optimised further to guarantee Pareto-stationarity, a prerequisite for local optimality [Ye22]. The three methods have different implications for the ultimate balance of client models. While both [Hu22b] and [Cui21] (in its pure form) guarantee that all clients improve during an aggregation step, only the latter considers the magnitude of gradients in the calculation. Thus, [Cui21] may force a greater balance between clients, to the potential detriment of overall performance in highly heterogeneous settings. In contrast, [She25] may sacrifice an outlier for the benefit of the system. Though undesirable to selfish clients, the latter could offer a defence against intentionally divergent updates submitted by a malicious client.

Multi-criteria aggregation. These algorithms perform aggregation based on multiple metrics that describe different characteristics of the federated system, such as the accuracy of the global model and fairness between clients. Such criteria are not generally differentiable with respect to the model, and thus cannot be optimised using gradient-based methods [Kan24a]. Solution approaches rely instead on heuristic insights or the formulation of the aggregation step into a mathematically solvable optimisation problem.
[Meh22] propose an algorithm that can incorporate multiple arbitrary system objectives, including fairness metrics, on the server. Aggregation is accomplished by assigning weighted ranking scores to each client for its contribution to optimising each objective, calculated using a validation dataset possessed by the server. These scores are used to compute aggregation weights. In contrast, [Ju24] formulate fairness-controlled FL as a dynamic multi-objective problem, where the optimisation problem consists of a linear combination of client losses, with weights adjusted dynamically to balance the progress of all component objectives. This approach yields different trade-off solutions between fairness and convergence depending on the value chosen for a fairness parameter. The idea of optimising a weighted linear combination of objectives in the federated aggregation step was proposed before in [Li20b], generalising ideas from [Moh19]; but neither work explicitly acknowledges a multi-objective view of the problem. Both aggregation strategies have different strengths and weaknesses. [Meh22] offers transparent server-side evaluation of clients, including the potential to automatically recognise low-quality or malicious clients. However, the need for a validation dataset on the server may violate the privacy requirements of clients, and renders the method vulnerable to data poisoning attacks. Conversely, [Ju24] offers mathematical fairness guarantees, but little transparency in the aggregation process. In

addition, this algorithm may be vulnerable to malicious client participation.

2.2.2.2 | Online multi-objective hyperparameter optimisation

Algorithms that use MOO to optimise hyperparameters for the federated system may run off-line or on-line. In on-line algorithms, the optimisation process is integrated into the federated algorithm, i.e. parameters are changed during the runtime of the FL process. On-line candidate generation is typically integrated on the federated server at the aggregation step, with local training rounds used for evaluation. Existing works on online MO-HPO in FL can again be divided into hyperparameter tuning and neural architecture search.

Online hyperparameter tuning. The work by [Bad24] performs on-line hyperparameter optimisation for clients, generating and transmitting new parameters during each aggregation step. These parameters, a fairness constraint regularisation parameter and the learning rate designed to enforce fairness locally, are recomputed on the server-side by using multi-objective Bayesian optimisation. Finally, [Ban22] propose a multi-objective on-line device selection approach to speed up the learning process in the presence of stragglers. The selection algorithm is designed to maximise the available computing and communication resources on selected clients, using NSGA-II.

Online neural architecture search. NAS algorithms may be designed run on-line, modifying during the execution of the federated algorithm the structure of the neural network to be trained by each client. Such a strategy could significantly reduce the computational cost of the search, at the price of complicating the training and aggregation process by introducing dynamic parameters. The only such algorithm currently existing in the MOFL literature dynamically optimises the accuracy and evaluation speed of federated model training [Zhu22]. The NSGA-II algorithm is used during each aggregation step to generate partial samples of the full model to assign to clients for training. On-line MO-NAS presents a difficult challenge and is currently underexplored in the literature, but could offer significant efficiency benefits.

2.2.3 | Federated multi-objective learning

In federated multi-objective learning, the solving of a multi-objective learning problem (MOLP) is the ultimate goal, and FL acts as an auxiliary tool to facilitate learning in distribution. A major challenge compared to the class of MOFL algorithms is that in this setting, there is no control or information about the compatibility of the objectives involved in the problem, whereas in MOFL the objectives were designed to suit the federated setting. Note also that FL techniques have largely been developed for neural networks, so the focus in this setting is on MO-algorithms that train such models. Compared with the application of MO techniques to FL algorithms, the federated solving of MOLPs has received very little attention so far. Here we aim to offer a classification of the few existing works, and extrapolate the open challenges and problems that remain to be solved. See also Table 2.2 for a representative overview of existing works. On the most fundamental level, algorithms in this category can be separated by the number of solutions they are designed to find: one

Table 2.2: Comparison of selected Federated Multi-objective Learning (FMOL) algorithms. Note that all algorithms are dedicated to handling local multi-objective learning. As noted in Section 2.1.3, this requires modifications at several levels of the federated system.

Reference	Taxonomy label	Local MOO method	Global MOO method	Objectives
[Yan23b]	single-solution	successive single-obj. updates	MGDA	arbitrary
[Ask24]	single-solution	linearised objectives	MGDA	arbitrary
[Har23a]	server-led	linearised objectives	offline metaheuristic	arbitrary
[Sen24]	multi-task	multi-task layer	similarity-based partial aggregation	arbitrary separable tasks
[Har24]	preference-driven	linearised preferences	similarity-based aggregation+clustering	arbitrary

single solution to the MOLP, or multiple solutions representing different trade-offs between the underlying objectives.

2.2.3.1 | Methods finding a single solution

FMOL algorithms designed to find a single solution aim to find an arbitrary Pareto-stationary solution. The advantage of such approaches is a relatively quick convergence, e.g. by exploiting gradients to locate the nearest solution. The main disadvantage is a lack of control over which solution out of all possible ones is found, and thus a lack of choice for potential users. One of the earliest such works [Yan23b] once again extends the MGDA algorithm to the federated setting, this time with respect to client objectives. Local training sequentially updates client models with respect to each component objective. Then, clients submit a gradient vector for aggregation to the server, where MGDA yields optimal aggregation weights to update the global model. This algorithm is shown to converge to a Pareto-stationary solution. A subsequent work [Ask24] points out a risk of local drift in this approach, as well as a high communication load caused by transmitting separate gradient updates for all objectives. The algorithm proposed to mitigate these issues is also based on server-side MGDA, but clients reduce communication cost by transmitting a compressed matrix of all objective gradients. Local drift is avoided via a similar modification: client updates are computed from a linear combination of all objective gradients rather than a series of single-objective updates. Tackling a different use case, [Kin24] discuss data-driven MOO problems, where a federated server attempts to solve a multi-objective problem, e.g. clustering, using only indirect information from distributed clients. In this unsupervised setting, no gradient-based strategies are possible; the server instead utilises a MOEA to solve the problem.

2.2.3.2 | Methods finding multiple solutions

Federated algorithms designed to find multiple solutions have one of two goals: they either attempt (1) to find a full Pareto front, i.e. a set of trade-off solutions, or (2) to find a personalised model for each participant. For both variants, participants may have different preferences over the same objective functions, or may even be solving entirely disjoint tasks.

Finding a Pareto front. Algorithms that aim to find a Pareto front of solutions must explore a wide range of the search space to identify a diverse spread of trade-off solutions. In the distributed setting, this may happen at different levels of the federated system: *server-led* exploration sees the federated server managing the exploration and constructing a Pareto front. A first framework for such a scenario has been proposed in [Har23a], utilising a metaheuristic on the federated server to decompose the multi-objective problem in into single-objective candidate subproblems. This approach bears similarities to some of the top-level algorithms discussed in Section 2.2.2.2, in that each candidate is evaluated separately by a full federated system. Unlike those approaches, however, the full system is not strictly required for an effective evaluation. Thus, the efficiency of the evaluation could be improved by the use of an algorithm that can federate candidates with different objective preferences. To the best of our knowledge, such an algorithm has not yet been proposed in the literature. Future contributions may be able to leverage client-specific solution algorithms in combination with server-led Pareto exploration strategies.
In contrast, *client-led* exploration would have each client attempting to find a Pareto front, e.g. in cases where the server is untrusted or lacks computing resources. This scenario has, to the best of our knowledge, not yet been addressed in the literature, but would carry its own challenges and opportunities inherent to the federated setting, most importantly a shift of control from server to clients, and the alignment of local Pareto fronts. Possibly related is the *fully-distributed* setting, where no server is involved in the training process and aggregation is decentralised across the client network.

Finding client-specific solutions. Here, the goal of the algorithm is to find a solution for each client in the system, based on different local requirements. Crucially, and in contrast to single-solution algorithms, this approach yields a different model for each client, matching that client's objectives, instead of finding a global model that generalises over all clients. This variant is known as Personalised FL, and is typically used in highly heterogeneous settings where the focus is on individual client performance [Tan23]. Note that this type of algorithm is arguably unique to the federated setting, arising from its properties that participants in FL are heterogeneous and may have different, independent interests.

In a *preference-driven* setting, client heterogeneity is induced by different preference weights assigned by each client to the same underlying multi-objective problem [Har24]. Formally, the objectives of the i-th client are weighted by that client's unique preference weights w^i:

$$\vec{f}^i(x) := \vec{w}^i \odot \vec{f}(x) = (w_1^i f_1(x), \dots, w_n^i f_n(x))^T \tag{2.5}$$

Where objective components are conflicting, learning trajectories of clients could diverge even on the same underlying model; the PFL approach is intended to embrace this diversity instead of counteracting it. Only a handful of works so far have considered a personalised approach to objective heterogeneity. In the first such work [Har24], client preferences are assumed to be private, and local training is performed on a weighted linear combination of the objectives. The challenge in this setting is to aggregate clients whose current training trajectory is compatible, and separate clients where it is not. As little direct information

about the mutual compatibility of clients is available on the server, many classical MOO methods cannot be applied. Instead, the proposed algorithm performs clustering and weighted aggregation based on the similarity of model updates.
Federated *multi-task* learning is an edge case scenario where clients solve mutually different subsets of tasks (i.e. objectives). A number of works in the FL literature, e.g. [Gho20] and [Hua23], have addressed a simplified setting where each client is assigned a single task[1] without acknowledging a multi-objective perspective. To the best of our knowledge, only one work currently considers the problem where each client is assigned a set of several tasks [Sen24]. Similarly to other works on FMOL, this task assignment is private. Under the proposed algorithm, clients jointly train a block of shared model parameters plus a separate parallel model layer for each task to be solved by the client. Once again, clients are aggregated based on a model similarity score, computed here based both on the shared parameters and a matching of task-specific layers.

2.3 | Conclusion and perspectives

In this chapter, we have presented the first comprehensive survey on the use of multi-objective methods in connection with Federated Learning. In Section 2.2, we have proposed a novel taxonomy to classify existing works in the literature, based on the three layers of the federated system. For each category, we have offered a perspective on recent trends, open challenges and possible approaches. Existing work demonstrates that MOO is a promising tool to improve transparency and effectiveness of FL techniques when navigating real-world problems. As in the wider field of FL, further work remains to be done. Open avenues of research in MOFL include, most prominently, (i) effective defence against malicious attackers in multi-objective aggregation; (ii) the use of MOO methods specifically to recognise low-quality clients; (iii) enhancing transparency and control of MO-preferences for users, e.g. by generating multiple different Pareto-optimal solutions, and (iv) exploring more sophisticated MOO techniques, e.g. to replace the baseline NSGA-II algorithm that is currently used in many of the works discussed here. The area of FMOL, enabling the federated solving of multi-objective learning problems, remains largely open. Initial contributions to the field could include, for example, (v) improving the efficiency of server-led strategies finding a Pareto front; (vi) exploring the effect of preference heterogeneity on convergence in single- and multi-solution algorithms; (vii) exploring the cumulative effect of data heterogeneity on FMOL problems; (viii) considering variant FMOL settings, e.g. where client preferences are not private.

In the following chapters, we discuss our algorithmic contributions to the area of Federated Multi-objective Learning in more detail, beginning with with the MOFL/D framework.

1 Note that the 'multi-task' label is assigned inconsistently in the existing FL literature, referring variously to clients with heterogeneous datasets or objectives.

3 | MOFL/D: A framework for federated multi-objective learning

Contents

In this chapter, we present a first systematic treatment of the Pareto-based multi-solution branch of federated multi-objective learning, as defined in the taxonomy in the previous chapter. Pareto-based approaches typically tackle a federated setting where user (client) preferences are not yet defined at computing time. In the domain of multi-objective optimisation, the typical solution approach to such problems is to find a set of optimal trade-off solutions for a user to choose at a later time. This first general framework combines elements of federated learning (FL) and multi-objective optimisation (MOO), specifically multi-objective optimisation with decomposition (MOO/D).
Significant previous research exists on the topic of multi-objective machine learning (MOML) in the centralised setting, with a large majority of contributions focused on optimising the hyperparameters of a machine learning algorithm alongside an underlying single-objective problem[Mor23][Ale19][Súk22]. Other works tackle the extension of specific algorithms to the multi-objective case, e.g. [Liu21] and [Yan23a]. However, despite the prevalence of such problems and the existing research on MOML, there appears to be no previous research on the integration of multi-objective learning into the FL paradigm. Therefore, we begin this direction of research by formulating a framework that utilises concepts from the field of MOO itself, to allow a later systematic integration of existing approaches from related

fields.
The problem of multi-objective optimisation has been studied for decades [Sha22]. Problems can be classified into those where user preferences are known at the time of optimisation, providing an ordering between objectives, known as *a priori* problems, and *a posteriori* problems, where preferences are unknown. One approach of solving such multi-objective problems is by decomposition (MOO/D) – a common decomposition method is to scalarise the set of objectives to obtain a single-objective problem, with different scalarisations producing different subproblems. Here we choose a linear scalarisation approach.

As discussed in Chapter 2, some previous literature exists on the application of multi-objective concepts to federated learning. This contribution falls into the taxonomy branch of *federated multi-objective learning*, the lesser-studied branch of the taxonomy proposed there. At the time of publication, the work by H. Yang et al. [Yan23b] was, to the best of our knowledge, the only other existing work to tackle federated multi-objective learning in a general way. This work differs from MOFL/D in several important respects: First, the framework proposed there is a *single-solution FMOL* approach, designed to find only a single global solution to the multi-objective problem. In contrast, our work is a *server-led multi-solution approach*, generating a Pareto front of solutions representing different trade-offs between the objectives. This approach allows for the later selection of solutions based on different priorities without the need to recompute. Second, their setting assumes that the knowledge of each client is permanently limited to a subset of all relevant objectives. In our work, we assume that all objectives are known to all clients, and that clients are capable of modifying their preferences over these objectives. Third, their framework is based on multi-gradient descent, whereas we rely on a decomposition approach.

The remainder of this chapter is organised as follows: Section 3.1 introduces the formulation of the MOFL/D framework. In Section 3.2, a possible instantiation of this framework is demonstrated and its performance is validated on a number of multi-objective variants of well-established single-objective benchmarks. Section 3.3 contains a summary of this work.

3.1 | Description of the MOFL/D framework

In this section, we first introduce and formalise relevant concepts from Federated Learning and multi-objective optimisation; then we present the general MOFL/D framework.

3.1.1 | Background

In the FL setting, a set of n training samples $\mathcal{P}$ is partitioned into m subsets $\mathcal{P}_1, \dots \mathcal{P}_m$, with each $\mathcal{P}_i$ privately owned by a client $\mathcal{C}_i$. Each dataset cannot be shared outside of the client that owns it. Let $|\mathcal{P}_i| = n_i$ be the size of the i-th training set. In this work, we consider the classical horizontal FL setting as defined in [Yan19], where all clients observe the same features and client model architectures are homogeneous. Though classical Federated Learning [McM17b] was formulated to learn a global model θ that optimises a single objective function, here we assume instead that each client is optimising a *vector* of objectives $\vec{f}_i$. In the spirit of the assumptions made in the horizontal FL setting, we assume that all m clients optimise the same set of objectives; so the equivalent formulation

of the classical FL problem becomes

$$\vec{f}(\theta) = \sum_{i=0}^{m} \frac{n_i}{n} \vec{F}_i(\theta), \text{ where } \vec{F}_i(\theta) := \frac{1}{n_i} \sum_{p \in \mathcal{P}_i} \vec{f}_p(\theta). \tag{3.1}$$

Recall that in the absence of a pre-defined hierarchy of objectives, the set of solutions to this problem is a partially ordered set, as the value of different objectives is not comparable in terms of overall optimality. In such cases, with preferences unknown during the optimisation process, a common MOO approach is to find a set of solutions, each representing an optimal trade-off between objectives. We say that a solution v *Pareto dominates* another solution u iff it improves the value of at least one objective while matching or improving all others. In formal terms, we hold for a maximisation problem:

$$v \succ_p u \iff \exists i : f_i(v) > f_i(u) \wedge \forall j : f_j(v) \geq f_j(u).$$

The *Pareto front* $\mathcal{PF}$ of a set of solutions $\mathcal{S}$ is then defined as the subset of all solutions that are not Pareto dominated by any other solution:

$$\mathcal{PF}(\mathcal{S}) := \{ v \in \mathcal{S} \mid \neg \exists u \in \mathcal{S} : u \succ_p v \}.$$

3.1.2 | The MOFL/D framework

The overall goal of our MOFL/D framework is to find a set $\mathcal{M}$ of solution models, using the federated system, that together approximate the Pareto front of the objective space. In abstract terms, this may be modelled as shown in Fig. 3.1: A federated system consists of multiple participants, coordinated by a server, with each participant learning to optimise

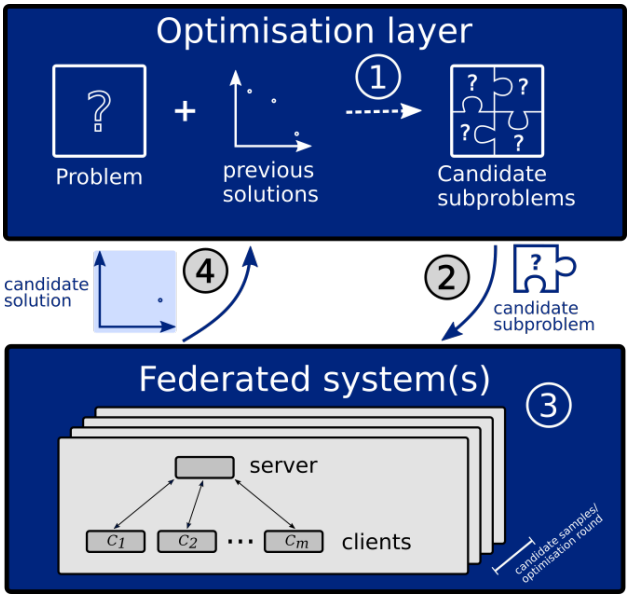


Figure 3.1: A high-level depiction of the theoretical MOFL/D framework.

an multi-objective learning problem as defined in Eq. 3.1, using a given scalarisation. An optimisation layer is added on top and given control of the federated system in order to manage the overall optimisation process. This optimisation layer carries out three tasks: (i) decomposing the MOLP into candidate sub-problems by generating scalarisation weights, (ii) managing the federated system to compute candidate solutions to each scalarisation provided by the optimisation layer and (iii) maintaining a set of optimal solutions out of the candidate solutions returned by the federated system.
At the beginning of each round, the optimisation layer generates a set of scalarisation weights to map the multi-objective problem to single-objective subproblems (step 1 in Fig. 3.1). The choice of candidate weights is governed by a metaheuristic method, making inferences from the results of previous optimisation rounds. (Note that this framework places no restrictions on the choice of multi-objective solver; any suitable method from conventional MOO may be used as a drop-in replacement.)
To solve the candidate problems generated thus, the optimisation layer invokes the federated system. A candidate weight is passed to the federated system (step 2 in Fig. 3.1), which executes a full FL cycle, computing a candidate solution to the scalarised problem (step 3). Once the federated system converges, the resulting model is passed back to the optimisation layer (step 4). This process is repeated for all candidates. For the sake of simplicity, we take a naïve approach in this first work, re-initialising the entire federated system for each subproblem and solving all subproblems in sequence. However, the question of how to use the federated system more effectively is a natural next step to continue our research.
Finally, the optimisation layer updates the current set of Pareto-optimal solutions discovered thus far, incorporating the results obtained from this most recent candidate generation. Depending on the choice of metaheuristic, a separate set of 'generating solutions' may also be maintained and updated at this stage, used to generate new candidate solutions or base models for initialisation. This optimisation cycle is repeated until a termination condition defined by the metaheuristic is met.

In addition to this main approach of generating scalar weights, we also propose the possibility of generating an initial base model for each single-objective problem, used to warm-start the federated training process. Previous works [Sat19], [Ngu22] have shown that FL tolerates, and may benefit from, initialisation with a pre-trained model chosen with sufficient care. We suggest that a base model could be derived from the solution obtained for a previous subproblem that is 'sufficiently close' to the current problem - a straightforward approach to quantifying problem similarity in this framework is to use the distance between the respective scalar weights used to generate each subproblem.

3.1.2.1 | Practical considerations on the federated system

Translating the abstract MOFL/D framework into an implementation requires two practical choices with respect to the federated system. The first choice is the implementation of the optimisation layer. In the preceding theoretical discussion of the framework, we have treated the high-level optimisation aspects of the algorithm as a fully separate layer; however, we note that in practice the optimisation layer may be integrated with the server functionality of the FL system. A second point to consider is the evaluation of candidate solutions. In a classical federated system, training samples are typically only available

to clients; in this case the final evaluation of any candidate solution would need to be performed on the client-side. This approach has the advantages of preserving data privacy and spreading the computational load of evaluation. However, the resulting estimate may not be representative of the system if the distribution of client data is skewed, and the self-reporting of solution values places a level of trust in clients that may be exploited by a malicious participant. Another approach also taken by some previous works, e.g. [Meh22], is to require a validation dataset to be known to the server; we follow this approach in our demonstration of the framework.

Algorithm 2 MOFL/D

Input: Number of iterations n_i, number of samples n_s, number of federated clients n_c.

Pareto front $\mathcal{PF}_0 \leftarrow \{\}$
Pareto front models $\mathcal{PFM}_0 \leftarrow \{\}$
$t \leftarrow 0$
while $t < n_i$ **do**
    $\mathcal{W}_t \leftarrow$ generate n_s candidate weights
    $\mathcal{V}_t, \mathcal{M}_t \leftarrow \{\},\{\}$
    for each $w \in \mathcal{W}_c$ **do**
        $\theta_0^w \leftarrow$ generate initial candidate model ▷ *Train federated system to completion to obtain global model*
        $\theta^w \leftarrow$ run Fed-Server with θ^w, w
        $\vec{v} \leftarrow$ evaluate θ^w for all objectives
        append $\theta^w, \vec{v}$ to $\mathcal{M}_t, \mathcal{V}_t$
    end for
    $\mathcal{PF}_{t+1} \leftarrow \mathcal{PF}_t \cup \mathcal{V}_t$
    $\mathcal{PFM}_{t+1} \leftarrow$ models generating $\mathcal{PF}_{t+1}$
    $t \leftarrow t + 1$
end while

3.2 | Experiments

In this section, we demonstrate an experimental validation of our MOFL/D framework on a number of multi-objective reinforcement learning (MORL) problems. We begin by providing a brief overview of the state of the art in the field of federated reinforcement learning; then we detail our choices regarding the instantiation and implementation of the framework. Finally, we discuss the design of the experiments performed and analyse our results.

A number of recent works study the application of FL to single-objective reinforcement learning [Qi21]. Zhuo et al. [Zhu20b] propose an algorithm that learns a secondary model to approximate the Q-network values of all clients without exposing their true networks. In [Zha23], multiple clients with different fixed preferences perform federated learning to obtain a generalised critic for carrying out local actor-critic reinforcement learning. While this work is one of the few where each client attempts to optimise multiple objectives, the proposed algorithm does not yield a Pareto front. Each client joining the learning process

must train its own actor model from scratch. Furthermore, it is not clear how this approach to federalising the training would generalise to other types of RL or non-RL algorithms. Finally, Jin et al. [Jin22] propose two algorithms, QAvg and PAvg, that extend the vanilla federated averaging (FedAvg)[McM17c] for use with Q-networks and policy networks, respectively.

3.2.1 | MOFL/D Instantiation and implementation

Figure 3.2: Illustration of multi-objective reinforcement learning environments used for validation experiments. Left to right: MO-Lunar Lander, Deterministic Minecart, Deep-Sea Treasure.

Faced with a lack of standard multi-objective benchmarking problems for this class of problem, we choose to use a number of multi-objective reinforcement learning (MORL) environments as our validation problems. We reason that these represent an intuitive class of multi-objective problems with varying characteristics and complexity, are extensions of classical RL baselines, and are implemented in a well-documented set of Python libraries [Fel23], making them easy to reproduce. We choose three standard MORL environments: MO-Lunar Lander, Deterministic Minecart, and Deep-Sea Treasure, all illustrated in Fig. 3.2. For the local learning algorithm used by clients to solve these problems, the existing literature provides a straightforward FL algorithm for single-objective reinforcement learning problems [Jin22].
Where possible, we make choices that resemble as closely as possible the equivalent baselines commonly chosen for demonstrations in the respective field of research; otherwise we choose methods based on their simplicity and ease of reproduction. A comprehensive overview of instantiation choices and applicable libraries used in the implementation is given in Table 3.1. The complete set of parameters chosen for all experiments is reported in the appendix. Noting that few reference parameterisations for MORL algorithms exist in the

Table 3.1: Instantiation and implementation choices for the experimental validation of MOFL/D.

Component	Instantiation	Implementation resources
Federated Algorithm	DQNAvg [Jin22]	morl-baselines [Fel22] stable-baselines3 [Raf21]
Learning Problems	Deep-Sea Treasure (DST) [Vam11] Deterministic Minecart (DMC) [Abe19] Multi-objective Lunar Lander (MOLL)	mo-gymnasium [Fel23]
Metaheuristic	Pareto Simulated Annealing [Czy98]	None used

literature, we have, where available, tested parameterisations for the related single-objective problems from the rl-baselines3-zoo [Raf20] benchmarking project; however, these did not always prove suitable to the multi-objective extension of the problem. Where no suitable parameterisation could be derived from the literature, parameters were tuned manually.

3.2.2 | Experiment design

We focus on investigating the impact of varying parameters of the federated system on the overall performance of the framework. We run experiments on each environment with two, three, and five clients in federation. In addition, we run the algorithm with the same configuration on single-client systems with no communication to obtain a baseline performance of the non-federated system. We also investigate the impact of the duration of the local training phase in the federated system, comparing runs with a local training phase duration of $2000, 5000$ and 10000 iterations. Finally, we contrast the performance of the algorithm on a federated system using pre-trained models and a federated system following the conventional approach of training models from scratch. We repeat experiments multiple times for each parameter combination, using different random seeds. All experiments on two- and three-client systems are repeated ten times, with the number of runs reduced to five for five-client systems in deference to the high computational cost of these experiments. Detailed information about the choice of random seeds for all experiments may be found in the appendix.

A standard method in the field of multi-objective optimisation is to study the subset of optimal trade-off solutions, or *Pareto front*, found by the algorithm [Nga05]. Intuitively, given a set of multi-objective solutions $\mathcal{S}$, a point s lies on the Pareto front iff the value of one objective in s cannot be improved without reducing that of another.
Many metrics designed to measure characteristics of a multi-objective solution set have been proposed in the literature, with most focused on quantifying the *diversity* and *convergence* of solutions [Riq15]. The diversity of a solution set describes the distribution of solutions in space - it is often considered more desirable to find solutions that are different from one another, in order to present a greater range of options to an end user selecting among the different possible trade-offs. The notion of convergence refers to the closeness of the obtained solutions to the underlying 'true' Pareto front - the closer the better. A set of multi-objective solutions is generally considered to be of high quality iff it has both a high diversity and high convergence – only one of these characteristics is not sufficient, as illustrated in Fig. 3.3. A set of solutions with high diversity and low convergence may offer a large selection of trade-off solutions, but all solutions are far removed from optimality. Conversely, a set of solutions with high convergence and low diversity may contain solutions that are close to optimal, but fail to cover the range of possible trade-offs. Only a set of solutions with both high convergence and high diversity yields a full range of near-optimal trade-off solutions.
In practice, the notions of diversity and convergence are difficult to quantify for the general case, in part because the 'true' Pareto front is often unknown; various surrogate metrics have been proposed in the MOO literature.

We evaluate the performance of our framework using three common multi-objective metrics [Zit03]: the hypervolume defined by our non-dominated solution set, the sparsity

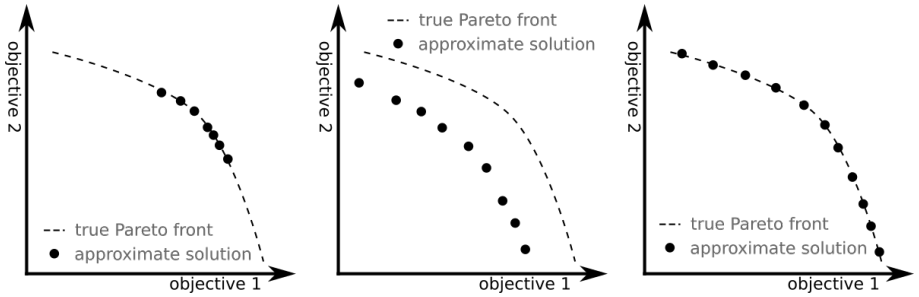


Figure 3.3: Illustration of diversity and convergence in a multi-objective context. Left: a solution set with good convergence and poor diversity; centre: poor convergence and good diversity; right: good convergence and good diversity. Both objectives are being maximised in these examples.

of the solution set [Xu20], and the inverted generational distance (IGD) [Coe05], using the set of all solutions to approximate the true Pareto front.

Hypervolume (↑)**.** The hypervolume metric is computed as the combined volume of the set of hypercubes spanned by the solutions on the Pareto front and a pre-defined minimal reference point. This metric captures both diversity and convergence: more diverse solutions on the Pareto front generate hypercubes with less mutual overlap, increasing the overall hypervolume, while more optimal individual solutions are further removed from the reference point, leading to a greater volume of their respective hypercubes. However, this metric suffers from some weaknesses that make it unfit to be used in isolation, e.g. small numbers of solutions that are near-optimal for a particular trade-off have the potential to dominate a more diverse set of different solutions. Therefore, a thorough analysis requires the use of other metrics in combination with the hypervolume metric.

Sparsity (↓)**.** The sparsity metric measures the mutual distance between solutions on the Pareto front; as such, it describes the diversity of a set of solutions. This metric, too, is limited when used in isolation: aside from not capturing convergence, it is also influenced by the number of solutions involved. A set of only two relatively close solutions will return a lower sparsity score than the same set with another, more distant solution added. This characteristic suggests the use of the cardinality metric to support a sparsity analysis.

Inverted Generational Distance (IGD) (↓)**.** This metric quantifies the convergence in terms of the distance of the Pareto front of the solution set to the 'true' Pareto front. As the true Pareto front of a problem is rarely known in practice, it is usually approximated as the Pareto front obtained when combining all solutions generated during an experimental campaign.

3.2.3 | Selected Results and Discussion

3.2.3.1 | Main experiments

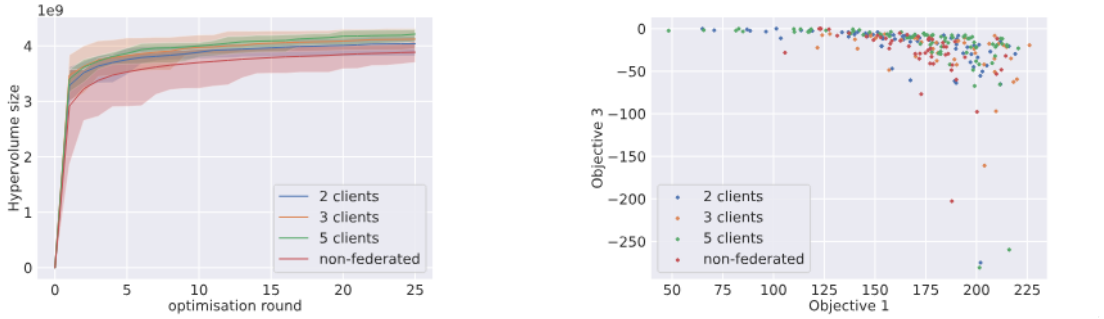
Numerical results are shown in Table 3.2. For all three learning environments, we consistently observe that the MOFL/D algorithm run with a federated system matches, and for the more complex problems outperforms, the same heuristic run with a non-federated system. This demonstrates both the general potential of federating the training of multi-

objective learning problems, and the validity of our framework. In more detailed terms, we observe a significantly increased hypervolume value along with a decreased sparsity in the results generated by running MOFL/D on multi-client systems, compared to a single client, on the two more complex MO-Lunar Lander (MOLL) and Deterministic Minecart (DMC) environments - see Fig. 3.4 for an example of the observed hypervolume evolution (Fig. 3.4(a) and associated solutions(Fig. 3.4(b)). The ultimate hypervolume values obtained for the Deep-Sea Treasure (DST) environment are similar for all federated systems and the non-federated system; this can likely be explained by the simplicity of the environment, with its very limited number of optimal solutions.

On the more complex environments we also observe a tendency for systems with a higher number of clients to find solution sets with greater hypervolume and lower sparsity. The impact of length of local training phase appears dependent on both the complexity of problem and number of clients in the federated system, with differing qualitative results for different environments. Finally, we observe no clear result on the benefits of re-using results to warm-start new training rounds: the ultimate performance of the system relative to non-pre-trained models differs across environments, with improvements in some and reduced performance in other cases.

3.2.3.2 | Impact of local training phase

We observe that the duration of the local training phase during federated learning has a notable impact on the overall performance of the MOFL/D algorithm. This matches previous experiences with optimising the performance of federated learning system outside of a higher-level framework. The optimal choice of the federated learning phase differs between the three experimental environments we consider, as is to be expected for problems of differing complexity. For the Lunar Lander environment, the longest tested local training phase (10000 iterations) ultimately produces the most optimal solution set, whereas shorter training phases tend to be more successful in the other two, less complex, environments



(a) Hypervolume evolution compared for systems with variable numbers of federated clients.

(b) Sample Pareto fronts obtained with the same configuration across variable numbers of clients.

Figure 3.4: Impact of different parameters of the federated system on hypervolume development for MOFL/D run on the MO-Lunar Lander environment. Experiments run with 10000 local steps per federated round and without pre-trained models.

Table 3.2: Numerical results of experiments on each benchmarking environment. Hypervolume and sparsity metrics are reported here; see Table 3.3 for the corresponding values of the IGD metric. Each entry reports the mean value of the respective metric, with the associated variance in parentheses. Higher hypervolume values and lower sparsity values, respectively, correspond to better performance.

Conf.	Hypervolume			Sparsity		
$n_c/n_i^f/ws$	DST	DMC	MOLL ($\cdot 10^{-7}$)	DST	DMC	MOLL ($\cdot 10^{-1}$)
$2/2k/T$	992.3(2.4)	896.8(33.4)	403.7(8.3)	17.9(3.0)	1.0(0.5)	353.5(634.3)
$2/2k/F$	970.8(39.9)	932.9(17.9)	404.6(8.2)	21.8(9.0)	1.5(2.9)	50.9(85.1)
$2/5k/T$	973.6(33.1)	867.6(58.2)	399.7(8.0)	22.3(11.8)	1.5(0.8)	95.6(166.0)
$2/5k/F$	990.8(3.3)	936.9(11.6)	405.2(11.3)	19.6(3.3)	0.5(0.2)	30.6(23.5)
$2/10k/T$	990.3(4.5)	854.6(58.1)	405.4(7.5)	19.8(3.9)	1.6(0.8)	141.8(338.8)
$2/10k/F$	985.8(14.7)	932.0(11.5)	404.0(10.7)	22.1(10.4)	0.5(0.2)	30.7(20.0)
$3/2k/T$	984.5(12.2)	869.0(60.0)	410.2(15.1)	26.1(12.8)	1.4(0.9)	108.3(247.2)
$3/2k/F$	986.6(10.7)	940.5(7.1)	405.2(8.3)	24.2(11.4)	0.4(0.1)	52.5(55.1)
$3/5k/T$	990.8(3.1)	893.6(49.7)	402.9(9.3)	20.0(4.9)	1.1(0.7)	210.1(565.0)
$3/5k/F$	974.8(40.0)	935.7(9.0)	406.0(6.0)	23.0(12.0)	0.5(0.1)	20.9(7.6)
$3/10k/T$	987.3(10.7)	819.2(44.8)	407.9(15.0)	22.0(7.6)	2.1(0.6)	104.2(134.0)
$3/10k/F$	**993.9** (1.3)	908.2(39.0)	412.3(11.7)	**15.8** (1.2)	0.9(0.6)	51.3(66.0)
$5/2k/T$	974.8(28.4)	908.0(1.9)	**425.0** (6.8)	46.1(46.1)	0.9(0.0)	68.9(104.2)
$5/2k/F$	988.2(9.1)	**941.1** (7.2)	408.1(6.7)	21.3(8.5)	**0.4** (0.1)	14.9(5.3)
$5/5k/T$	985.5(10.7)	890.0(47.1)	420.5(14.0)	27.4(13.9)	1.1(0.7)	57.4(67.1)
$5/5k/F$	991.4(2.6)	936.5(16.3)	411.2(11.7)	19.0(2.9)	0.5(0.2)	**12.9** (1.8)
$5/10k/T$	989.8(3.8)	886.4(44.8)	413.3(14.1)	22.1(5.8)	1.2(0.6)	207.4(276.3)
$5/10k/F$	992.1(3.3)	923.1(13.5)	421.4(7.8)	17.8(3.5)	0.7(0.2)	23.2(16.9)
Non-fed.	983.1(39.2)	879.8(73.3)	388.7(8.5)	35.7(116.9)	1.3(0.9)	108.5(147.9)

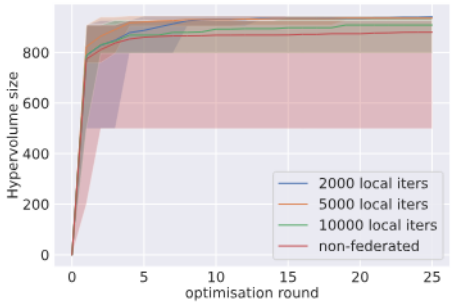
tested here. An inspection of the solutions obtained e.g. for the Lunar Lander environment clearly shows the impact of local training phase duration on the diversity of the solution set - see the projections of the solution sets shown in Figures 3.6(a), 3.6(b). The diversity of solutions obtained with a shorter local training phase is much lower for this environment, indicating that the federated system likely converges too quickly to a local optimum to adequately explore the solution space.

3.2.3.3 | Number of federated clients

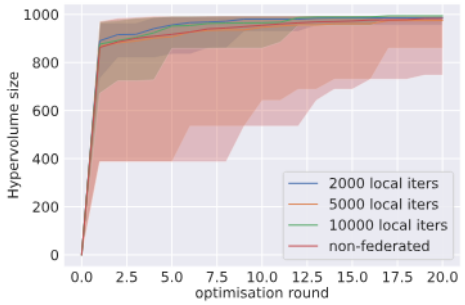
We observe that, in general, an increased number of federated clients leads to an increased performance of the MOFL/D algorithm - see e.g. the hypervolume evolution for the Deterministic Lunar Lander and Minecart, shown in Figures 3.4(a), 3.7(a); compare also Table 3.2. While this is not the case for the Deep-Sea Treasure environment (see Figure 3.7(b), a higher number of clients in this case still matches the performance of other systems. The lack of improvement for higher numbers of clients is very likely due to the limited complexity of the problem.

Table 3.3: Numerical results of experiments on each benchmarking environment. The IGD metric is reported here; for the corresponding hypervolume and sparsity values see Table 3.2. Each entry reports the mean value of the respective metric, with the variance in parentheses. Lower values of the IGD metric correspond to better performance.

Conf.	IGD		
$n_c/n_i^f/ws$	DST	DMC	MOLL
$2/2k/T$	0.113(0.1)	0.135(0.1)	24.729(2.0)
$2/2k/F$	0.393(0.5)	0.097(0.1)	24.030(1.7)
$2/5k/T$	0.399(0.5)	0.257(0.2)	25.017(2.1)
$2/5k/F$	0.189(0.2)	0.079(0.0)	24.693(1.6)
$2/10k/T$	0.218(0.2)	0.276(0.2)	23.850(2.4)
$2/10k/F$	0.287(0.4)	0.098(0.1)	24.290(2.5)
$3/2k/T$	0.426(0.5)	0.237(0.2)	24.452(4.0)
$3/2k/F$	0.367(0.5)	0.087(0.0)	23.617(2.1)
$3/5k/T$	0.189(0.2)	0.169(0.1)	22.834(2.0)
$3/5k/F$	0.412(0.5)	0.054(0.0)	23.351(1.8)
$3/10k/T$	0.273(0.3)	0.388(0.1)	23.920(2.5)
$3/10k/F$	**0.024** (0.1)	0.158(0.1)	22.019(1.8)
$5/2k/T$	0.748(1.0)	0.106(0.0)	22.411(2.7)
$5/2k/F$	0.271(0.4)	**0.054** (0.0)	22.138(1.5)
$5/5k/T$	0.366(0.4)	0.157(0.2)	22.462(2.5)
$5/5k/F$	0.151(0.1)	0.088(0.1)	22.780(2.1)
$5/10k/T$	0.308(0.3)	0.168(0.1)	21.549(2.3)
$5/10k/F$	0.082(0.1)	0.086(0.0)	**20.221** (1.5)
Non-fed.	0.335(1.1)	0.209(0.2)	27.912(2.0)

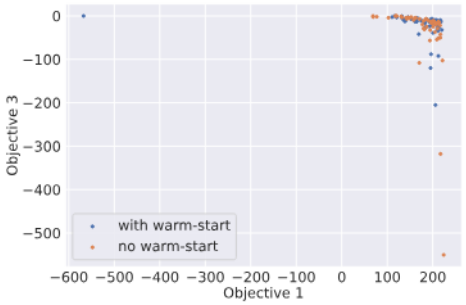


(a) Results for the Deterministic Minecart environment.

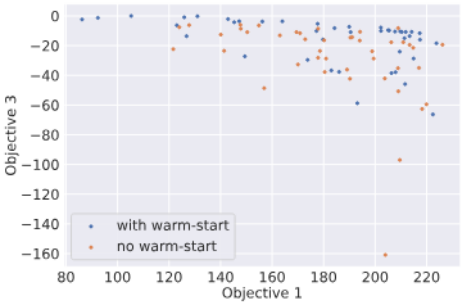


(b) Results for the Deep-Sea Treasure environment.

Figure 3.5: Hypervolume evolution compared for different durations of the local training phase in federated training. Experiments run with 3 federated clients and without pre-trained models.



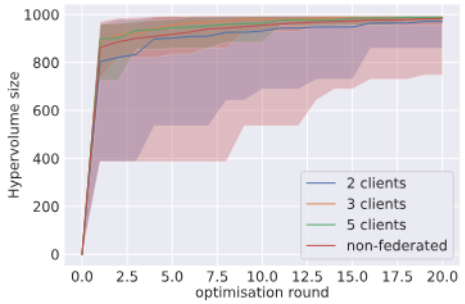
(a) Results for 2000 iterations per local training round.



(b) Results for 10000 iterations per local training round.

Figure 3.6: Solutions obtained for the Lunar Lander environment compared for different durations of the local training phase in federated training. Experiments run with 3 federated clients and without pre-trained models. (Solution vectors projected into the plane to show objectives 1 and 3. It is clear that the duration of the local training phase has a significant impact on solution diversity.

(a) Results for the Deterministic Minecart environment.



(b) Results for the Deep-Sea Treasure environment.

Figure 3.7: Hypervolume evolution compared for variable numbers of federated clients. Experiments run with 2000 local steps per federated round and without pre-trained models.

3.3 | Summary and Outlook

In this chapter, we have presented MOFL/D, a novel general framework to solve inherently multi-objective problems in a Federated Learning setting. The framework is designed to find a Pareto front of solutions for an arbitrary client-level multi-objective problem under the guidance of a server. Following the theoretical definition in Section 3.1, we have discussed instantiation choices for the framework and shown one such instantiation in Section 3.2. Using this instantiation, we have performed experiments on three well-founded benchmarking problems from the domain of multi-objective reinforcement learning, showing the validity of our framework and investigating the effect of several variable parameters related to the federated system.

The following chapter introduces an algorithm designed for a tackle a different multi-

solution FMOL use case.

4 | FedPref: personalised federated multi-objective learning under preference heterogeneity

Contents

This chapter presents an algorithm designed to solve a different setting in multi-solution federated multi-objective learning, where personalised preferences are known and fixed at computing time. The main challenge that arises from federating such a setting is the potential for heterogeneous objective preferences between the participating clients. To the best of our knowledge, there are no earlier works in the literature addressing this continuous objective-heterogeneous setting in detail. However, our problem is related to

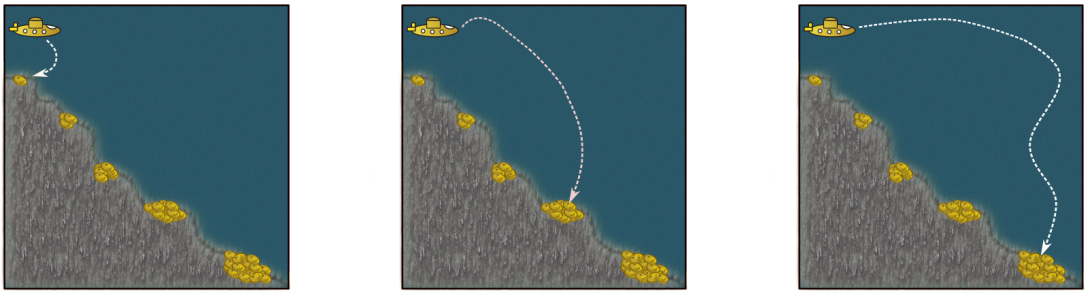


Figure 4.1: Different preferences lead to different solutions in a yellow submarine searching for underwater treasure. Left: A strong preference for minimising travel distance. Centre: Balanced preferences. Right: A strong preference for maximising the value of the treasure reward. The goal of our work is to allow clients with problems like these to perform FL effectively, despite their heterogeneous objective preferences.

other types of heterogeneity problems that present themselves in the FL setting, where the FL algorithm must account for differences between clients, such as data heterogeneity or hardware heterogeneity. Particularly in the case of data heterogeneity, client models also tend to develop in different directions – as is to be expected for clients in our objective-heterogeneous setting – making the comparison with our problem setting an interesting one. Many varied approaches have been proposed to address this problem [Ye23a]; these can be broadly divided by their approach to model aggregation [Tan23].
Some works follow the more classical approach of producing a single generalised global model, with the goal of adapting this model as well as possible to all individual client datasets simultaneously. One of the first such algorithms was the FedProx framework [Li18], which relies on regularisation to encourage model adaptation. To accomplish this, a new proximal term is added to the loss function of each client, penalising divergence of the local model from the global model. Other regularisation-based algorithms have since followed, e.g. [Kar20] and [Li21c], introducing variance reduction and model-contrastive learning, respectively.

In contrast, the goal of the federated aggregation in the second approach is to learn an individual model tailored to each client [Tan23]. This strategy is known as Personalised Federated Learning (PFL). Variants of PFL, in turn, may be separated into those based on a modified model architecture, such as parameter decoupling or knowledge distillation, and those based on model similarity-guided aggregation. Of the former approaches, knowledge distillation strategies can be costly, and the parameter-decoupling strategy may struggle to adapt to training with different objective functions. Therefore, we choose to place our focus on the latter approaches, as model similarity-based methods may be lightweight, appear to have the potential to adapt well to different types of heterogeneity, and require no additional information about clients. A number of such approaches have been proposed in the literature in recent years, e.g. [Lon22], [Gho20], [Dua21a]. In this work, we focus on two recent methods that appear most flexible, and so most likely to transfer well to the preference heterogeneous setting: the Clustered Federated Learning [Sat19] (CFL) algorithm, and

Many-Task Federated Learning [Cai23] (MaTFL).The former work, proposing the Clustered FL [12] (CFL) algorithm, is of particular interest here. It deals with settings where the underlying data distributions known to participants are not fully compatible, leading to conflicts in the training of a joint model. To solve this, the idea of CFL is to train clients together in a classical federation until the global model converges to a stationary point, allowing clients to learn from each other until mutual conflicts stall the training process. Then clients are permanently separated into clusters based on the similarity of model gradients in the stationary point. Our multi-objective preference-heterogeneous setting is related to the data-incongruity problem tackled by CFL, in that we expect clients with preferences for conflicting objectives to also produce incompatible models during training. However, the heterogeneity of clients may be more complex, given the number of potential objectives and different preference distributions. Therefore, we take inspiration from the clustering strategy of CFL for our approach, but additionally introduce the idea of personalising learning inside each cluster. Our aim is to allow a higher degree of individual exploration for clients at an earlier stage in the training process, without cutting off cooperation earlier than necessary. We propose FedPref, an algorithmic approach based on Personalised Federated Learning (PFL), where each federated client learns an individual model tailored to its needs, and different objective preferences lead to different solutions. The goal of our algorithm is twofold: first, to optimise individual client performance, as is common for other PFL approaches. However, we also want our algorithm to perform well under a multi-objective view of the federated system as a whole, conforming to the common expectations of multi-objective problem solving. In order to measure our success at this secondary goal, we introduce a novel analytical view of the federated system itself, using metrics common in the fields of multi-objective optimisation and multi-objective learning to assess the diversity and convergence of the set of solutions found by all clients.

The remaining content of this chapter is organised as follows: In Section 4.1, we motivate the need for a personalised federated learning approach with the aid of an intuitive example. In Section 4.2, we formalise the problem and introduce the FedPref algorithm designed to solve it. In Sections 4.3 and 4.4, we evaluate the performance of FedPref experimentally, first using standard FL metrics and then by applying multi-objective metrics to the federated system. Finally, Section 4.5 presents a summary of this chapter.

4.1 | Motivation for a personalised approach

In this section, we introduce the need for a personalised solution approach to the preference-heterogeneous setting in some detail, using an intuitive example to show how a single generalised solution would not satisfy the typical expectation of a preference-based multi-objective approach.

Performing federated aggregation with preference-heterogeneous clients can be challenging, as conflicting objectives may lead to substantially different models. In such a setting, the classical approach of training a single global model would struggle to satisfy different preference distributions simultaneously. Besides the technical difficulty of aligning heterogeneous clients, the non-personalised aggregation approach also fails to respect the underlying intention behind preference-guided learning. Generally, the idea behind

modelling a problem with multiple objectives is to allow multiple diverse solutions, with each representing a different trade-off between the various objectives. A single global solution, even one delivering high objective values for all involved clients, would not be a satisfying outcome in this scenario.

For an intuitive example in support of solution diversity, consider the scenario illustrated in Fig. 4.1. In this toy scenario, several submarines are searching for underwater treasure. Each has two separate objectives: minimising the diving distance, and maximising the haul of treasure recovered. We observe that these objectives are conflicting, as more valuable treasure is located deeper down on the seabed, necessitating a longer dive. In assigning different preference weights to the two objectives when solving this problem, we expect to recover different behaviours, as illustrated in the different panels of Fig. 4.1. In the left-most image, the submarine has a high preference weight placed on the travel-distance objective, and so travels to the closest treasure. The right-most image shows the reverse: the submarine has a high preference for finding treasure, and so dives as far as necessary to reach the most valuable location. This is the learning outcome a user might expect in assigning preference weights; yet a non-personalised federated learning algorithm might instead converge to the same solution for all clients, shown in the central illustration. This represents a "middle ground" between the preference distributions of the two others, potentially leading to comparable scalarised results for both. However, the different preference weights have essentially lost their expected meaning: the user has no perceived control over the learning outcome.

4.2 | The FedPref algorithm

In this section, we define the FedPref algorithm and relevant concepts. We begin by formally defining the problem setting in Section 4.2.1, followed by an initial sketch of the algorithm and a definition of the underlying similarity metric in Section 4.2.2. Finally, we discuss the components of the algorithm in more detail: the weighted aggregation strategy is described in Section 4.2.3, the clustering strategy in Section 4.2.4, and the full FedPref algorithm in Section 4.2.5.

4.2.1 | Problem formulation

We want to perform personalised Federated Learning across n clients, each of which has a learning problem with m distinct objectives $f_1, \cdots, f_m$. There is no general importance order assigned between objectives, but each client has a personal fixed preference weight vector across all objectives. See also Fig. 4.2 for an illustration of the problem and the federated learning process.

Following a classical approach in multi-objective optimisation[Sha22], we map this multi-objective problem to a single-objective problem in order to solve it, so that all clients learn a linear combination of these same objectives, with the preference weights assigned as scalars. So client i, with preference distribution $\vec{w^i} = (w_1^i, \dots, w_m^i)^T$, is optimising the objective function

$$f^i(\theta) = f(\vec{w^i}, \theta) = \vec{w^i}\vec{f}(\theta) = \sum_j^m w_j^i f_j(\theta). \tag{4.1}$$

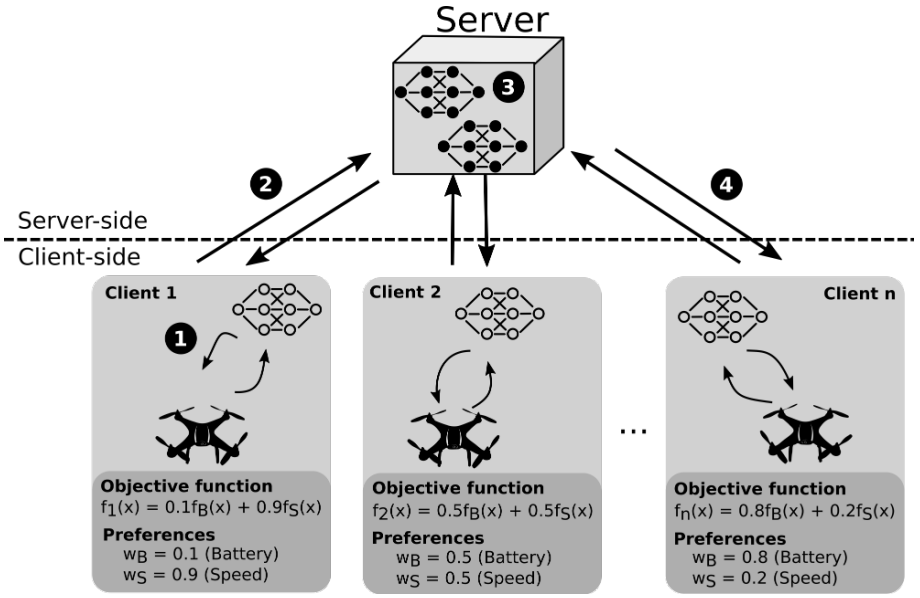


Figure 4.2: An illustration of the federated system solving a multi-objective problem with personalisation. In this instance, we want to learn to plan trajectories for drones, under two potentially conflicting objectives: conserving energy and maximising speed. Each drone assigns different importance (preference weights) to these objectives. Federated Learning takes place as follows: (1) Clients (drones) perform local training, using the objective function defined by their preferences. (2) Clients submit model updates to the server. (3) The server aggregates these model updates, obtaining personalised models. (4) The server returns the respective personalised models to the clients.

The preference distribution of each client is unknown to all other participants, including the federated server. (We can assume without loss of generality that all single-objective components f_j are known to all clients.) Each client i trains a *personalised* model θ_i using its personal preference weights. A major challenge in this scenario is that the objectives of federated clients may conflict, and these conflicts can lead to the divergence of client models at any stage of the training.

4.2.2 | Concept sketch and definitions

The fundamental idea of FedPref is to combine a recursive clustering mechanism, similar to [Sat19], and an adaptive weighted aggregation scheme, both based on a model similarity metric. The underlying idea behind this combination is to enable effective grouping and aggregation of clients whose preferences are compatible during the learning process (provided by the clustering component), while also maintaining the flexibility of training a personalised model for each client using weighted aggregation. Compare Fig. 4.3 for a visual representation of the flow between these components. Initially, all participating clients are grouped together in a single cluster. During every aggregation step, a personalised model is computed for each client, using adaptive weights computed based on mutual model similarity between pairs of clients. The mean model of all clients in the cluster serves as an indicator of the success of the intra-cluster collaboration: the mean model converges if

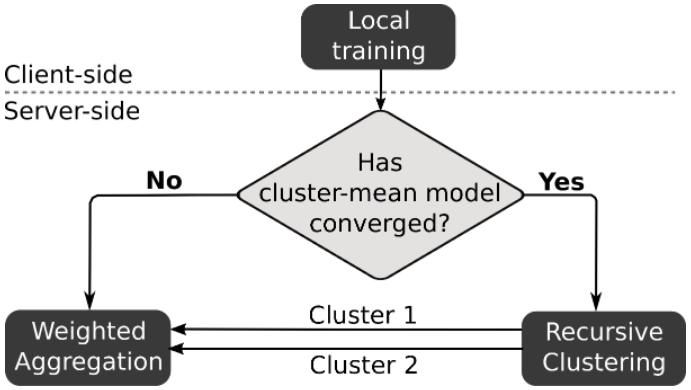


Figure 4.3: A schematic representation of the flow between components of the algorithm.

either all clients converge, or if the gradients of personalised client models start developing in conflicting directions. In this case, we perform a recursive clustering step, splitting the current cluster in two based on the same mutual model similarity metric that is used for the weighted aggregation. The learning process is then continued in the same manner inside the new clusters.

4.2.2.1 | Similarity metric

Before discussing the functionality of each component in detail in the following sections, we shall formally introduce the modified similarity metric that underpins both components. The similarity metric in aggregation round t is computed on the basis of model updates

$$\Delta\theta_i = \theta_i - \bar{\theta}_C^{t-1}, \tag{4.2}$$

where $\bar{\theta}_C^{t-1}$ is the cluster-mean model obtained after the previous aggregation step. Using these gradients, we define the similarity metric $sim(\cdot,\cdot)$ of two models θ_i and θ_j as

$$sim(\Delta\theta_i, \Delta\theta_j) = \frac{1}{L}\sum_{\ell}^{L} cossim(topR(\Delta\theta_i^{\ell}), topR(\Delta\theta_j^{\ell})), \tag{4.3}$$

where $\Delta\theta_i^{\ell}$ is the ℓ-th layer of $\Delta\theta_i$ and L is the total number of layers per model. The $topR$ operator is a variant of $topk$, where k is determined by the dimension of the input vector and a ratio $R \in (0,1]$. $TopR$ maps a vector $\vec{v}$ to a vector of the same dimension where the top $k = \lceil dim(\vec{v}) \cdot R \rceil$ elements of $\vec{v}$ (in absolute terms) are retained and the remaining elements set to zero. So for $topR(\vec{v}) = \vec{u}$, we have

$$u_i = \begin{cases} v_i, & \text{if } |v_i| \text{ in top } \lceil R \cdot dim(\vec{v}) \rceil \text{ absolute elements of } \vec{v}. \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

The cosine similarity $cossim(\cdot,\cdot)$ is defined in the standard way:

$$cossim(\vec{u},\vec{v}) = \frac{\langle \vec{u},\vec{v} \rangle}{\|\vec{u}\| \cdot \|\vec{v}\|}. \tag{4.5}$$

The rationale for this modification lies in the geometric interpretation of the cosine similarity metric (illustrated in Fig. 4.4): recall that the plain cosine similarity describes the cosine of the angle between two given vectors, with a value of -1 equivalent to antiparallel vectors, a value of 0 denoting orthogonal vectors and the maximum value of 1 denoting parallel vectors. When applied to model gradient updates, this metric can then describe – quite intuitively – how similarly two models are developing. This insight is leveraged e.g. in the CFL algorithm [Sat19].
However, we note that complications can arise from this application of the plain metric to model updates, particularly relating to the potentially high dimensions of models and the choice of reference point. The former point rests on the observation that in high-dimensional spaces, the cosine similarity metric is affected by the "curse of dimensionality", rendering comparisons of high-dimensional vectors in dense spaces, such as the weights of a neural network, increasingly difficult. In other works, e.g. [Sat19][Cai23], this is mitigated to an extent by comparing the individual layers of models instead of the complete flattened model. However, with the trend towards larger and larger models, we attempt to find a more general solution by introducing the 'topR' filtering of layer-gradients. The intention of this step is to sparsify the space in which vectors are compared, in the hope of obtaining more meaningful results.
The latter complication is founded on the need for a reference point in defining the gradient vectors to be compared. As we want to train personalised models, including for clients inside the same cluster, models do not begin each local training round with a common model (as would be the case in e.g. FedAvg, or CFL). Therefore, we need to explicitly define a model to compare to, ideally one that is both close to each client's actual model and whose difference accurately represents the relation of the clients being compared. We choose the cluster-mean model, obtained after the aggregation of the previous round has concluded, as this reference point for our algorithm.

To recapitulate, we choose to use this modified metric instead of the more common direct applications of cosine similarity for two main reasons:

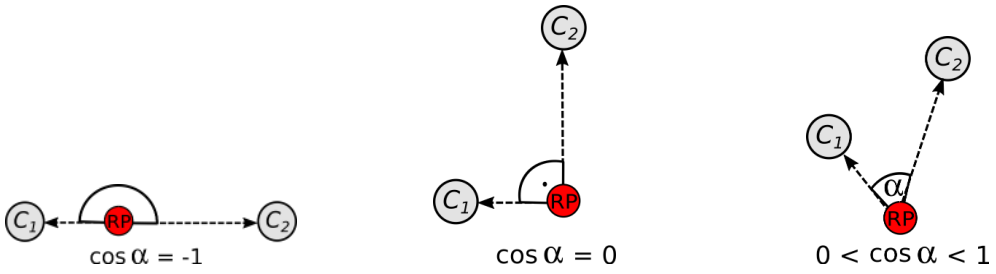


Figure 4.4: Geometric interpretation of cosine similarity.

- We hope to mitigate the "curse of dimensionality" that makes this metric increasingly meaningless for larger vector dimensions.
- Selecting the subset of the largest weights for each layer allows us to compare the most impactful, or "important" aspects of the models. This could lead to more meaningful decisions about which models to aggregate together.

We will show in Section 4.3 that, when compared to the use of the pure cosine similarity metric without weight selection, the use of this metric does indeed lead to improved results in our validation experiments.

4.2.3 | Weighted aggregation

The weighted aggregation – described in the pseudocode in Alg. 3 and illustrated in Fig. 4.5 – is carried out by the server for each separate cluster. For each cluster, the weighted aggregation phase begins with computing the similarity matrix of all clients contained in the cluster. The similarity metric (defined in Equation 4.3) returns a value between -1, representing the lowest possible mutual similarity, and $+1$, representing the highest possible similarity. These values are then clipped to a minimum lower similarity bound s_{min} – given to the algorithm as a parameter during initialisation – and subsequently normalised to the range [0,1] (see line 5 in Alg. 3). This step can be used to enforce a minimum similarity required for aggregation, as it essentially excludes all clients whose similarity to another client is lower than the given threshold from the aggregation with that client. Following this precomputing of similarity values, the actual personalised aggregation takes place: to compute the new personalised model for each client, the row corresponding to this client in the similarity matrix is taken as aggregation weight vector, normalised once more so that the sum of weights adds up to one, and finally used to compute the weighted average of all client models – see lines 9 and 10 in Alg. 3. This aggregation is carried out for each client inside the cluster; then the resulting personalised models are returned to the respective clients.

Figure 4.5: A weighted aggregation step inside a single cluster. Left: personalised client updates are computed using aggregation weights based on client similarity relative to the cluster-mean. Right: The updated cluster-mean is computed.

Algorithm 3 Weighted aggregation

C list of c clients in cluster
$\mathcal{W} \leftarrow (0)^{c \times c}$ ▷ Init aggregation-weight matrix
for $i \in C$ **do**
  for $j \in C$ **do**
    $w_{ij} \leftarrow (sim(\Delta\theta_i, \Delta\theta_j) - s_{min})/(1 - s_{min})$
  end for
end for
for $i \in C$ **do**
  $\hat{w}_i \leftarrow w_i/|w_i|$
  $\theta_i \leftarrow \sum_{c \in C} \hat{w}_{ic}\theta_c$
end for
return $(\theta_c | c \in C)$

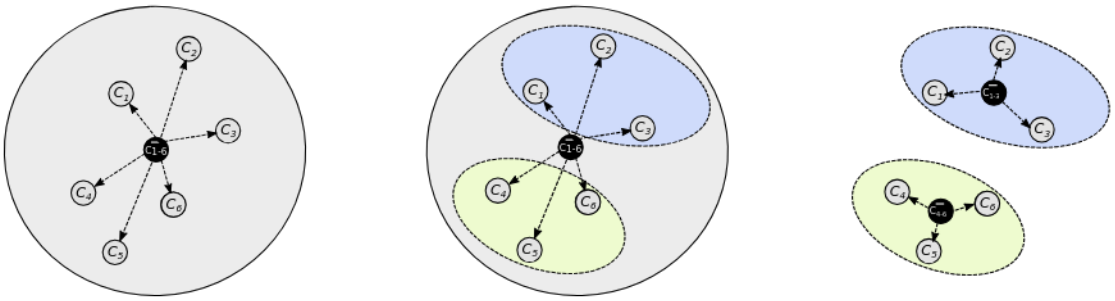


Figure 4.6: A recursive clustering step. Left: A cluster with cluster-mean at the beginning of an aggregation round. Centre: Clients inside the cluster are split into two clusters based on pairwise similarity relative to cluster-mean. Right: The resulting two clusters with respective cluster-means.

4.2.4 | Recursive clustering

The clustering procedure, illustrated in Fig. 4.6, is performed whenever a cluster is found to have converged during an aggregation round, i.e. where the clients inside the cluster no longer benefit from federated collaboration – the exact convergence criterion is discussed in Section 4.2.5. The purpose of this procedure is to separate the clients contained in the cluster into two new sub-clusters in such a way that clients whose models are developing similarly are grouped together to continue learning from each other, and clients that are developing in different directions are separated. This bipartitioning is based on the same similarity metric (Equation 4.3) as is used for the weighted aggregation. In principle, any suitable clustering algorithm can be utilised to perform the clustering itself; in this work, we choose to use spectral clustering [Dam18], as it tends to produce well-balanced clusters, performs well for low numbers of clusters, and an implementation is readily available in common libraries. The clustering procedure is performed no more than once per cluster per aggregation round.

Algorithm 4 Clustering

Require: $\|\Delta\bar{\theta}_C\| \leq \varepsilon$ ▷ Cluster-avg model change $\leq \varepsilon$

1: C list of c clients in cluster
2: $\mathcal{S} \leftarrow (0)^{c\times c}$ ▷ Init similarity matrix
3: **for** $i \in C$ **do**
4: **for** $j \in C$ **do**
5: $\Delta\theta_i, \Delta\theta_j \leftarrow \bar{\theta}_C^{t-1} - \theta_i, \bar{\theta}_C^{t-1} - \theta_j$
6: $\mathcal{S}_{ij} \leftarrow sim(\Delta\theta_i, \Delta\theta_j)$
7: **end for**
8: **end for**
9: $C_1, C_2 \leftarrow SpectralClustering(C, \mathcal{S}, 2)$ ▷ Bipartition $\mathcal{C}$
10: **return** C_1, C_2

4.2.5 | Full algorithm

The complete algorithm combines the weighted aggregation and clustering components, as detailed in Alg. 5 and conceptually in Fig. 4.3. In every round, all local models are trained for a fixed number of steps. Once all models for a given cluster C have been reported to the server, the aggregation phase begins. As a first step, the clustering criterion is checked: the difference of the cluster-mean model $\Delta\bar{\theta}_C$ of the most recent local updates to the cluster-mean model $\bar{\theta}_C^{t-1}$ following the latest aggregation round is computed (see lines $7-8$ in Alg. 5). If the magnitude of this change is less than a given convergence threshold ϵ, we assume that the models of clients inside the cluster are diverging. We therefore trigger the clustering process to bipartition the current cluster C into two new clusters C_1 and C_2. We then carry out weighted aggregation according to Alg. 3 on the new clusters, before updating the server-side record of current clusters.
If the clustering criterion is not met, aggregation continues in the preexisting cluster: weighted aggregation is carried out in this cluster, and client-membership of this cluster is recorded unchanged.
In one full server-side aggregation step, this procedure is executed for every cluster, with personal aggregated models returned to the clients of each cluster after aggregation has concluded. The algorithm terminates after T such aggregation rounds. Note that even if clients are still part of a larger cluster after $T-1$ aggregation rounds, the last aggregation step can be skipped after the final local training round, to allow clients a degree of local fine-tuning (see line 23 in the algorithm). We call this fine-tuning variant of the algorithm FedPref+FT, and the version without a fine-tuning step FedPref-FT.

Algorithm 5 NewFL-Server

$\mathcal{C} \leftarrow \{[1, \dots, n]\}$ ▷ Initial cluster
$\theta_1^0, \dots, \theta_n^0 \leftarrow$ Initialise client models
for $t \in 1, \dots, T-1$ **do**
  $\theta'_1, \dots, \theta'_n \leftarrow$ Train local models
  for $C \in \mathcal{C}$ **do**
    $\mathcal{C}_{temp} \leftarrow \{\}$
    $\bar{\theta}_C^{t-1} \leftarrow 1/|C| \sum_{c \in C} \theta_c^{t-1}$
    $\Delta\bar{\theta}_C \leftarrow \bar{\theta}_C^{t-1} - 1/|C| \sum_{c \in C} \theta'_c$
    if $\|\Delta\bar{\theta}_C\| \leq \varepsilon$ **then** ▷ Cluster converged
      $C_1, C_2 \leftarrow Clustering(\bar{\theta}_C^{t-1}, [\theta'_c | c \in C])$
      $\theta'_{C_1}, \theta'_{C_2} \leftarrow \{\theta_c^t | c \in C_1\}, \{\theta_c^t | c \in C_2\}$
      $\theta_{C_1}^t \leftarrow WeightedAggregation(\bar{\theta}_{C_1}^{t-1}, \theta'_{C_1})$
      $\theta_{C_2}^t \leftarrow WeightedAggregation(\bar{\theta}_{C_2}^{t-1}, \theta'_{C_2})$
      $\mathcal{C}_{temp} \leftarrow C_{temp} \cup \{C_1, C_2\}$
    else
      $\theta'_C \leftarrow \{\theta'_c | c \in C\}$
      $\theta_C^t \leftarrow WeightedAggregation(\bar{\theta}_C^{t-1}, \theta'_C)$
      $\mathcal{C}_{temp} \leftarrow \mathcal{C}_{temp} \cup \{\mathcal{C}\}$
    end if
  end for
  $\mathcal{C} \leftarrow \mathcal{C}_{temp}$
end for
$\theta'_1, \dots, \theta'_n \leftarrow$ Train local models ▷ Optional fine-tuning, replacing last aggregation round

4.3 | Client-level evaluation

In this section, we present a thorough experimental evaluation of our algorithm. We begin by introducing the general design of our experiments in Section 4.3.1, describing the problems and baselines we have selected for evaluation. In Section 4.3.2, we show and discuss the first part of our main validation experiments, evaluating the performance of our algorithm with a focus on average client performance under objective heterogeneity. Following this section, we introduce a multi-objective view of this problem setting in Section 4.4, giving a brief overview of common metrics, and analysing the performance of our algorithm with respect to these metrics. These experiments are supplemented by studies of specific characteristics of the FedPref algorithm: in Section 4.3.3, we perform an ablation study, comparing the individual performance of the clustering and weighted aggregation components with the combined algorithm; in Section 4.3.4, we analyse the sensitivity of the algorithm to two crucial parameters, and in Section 4.3.5 we evaluate the clustering strategy.

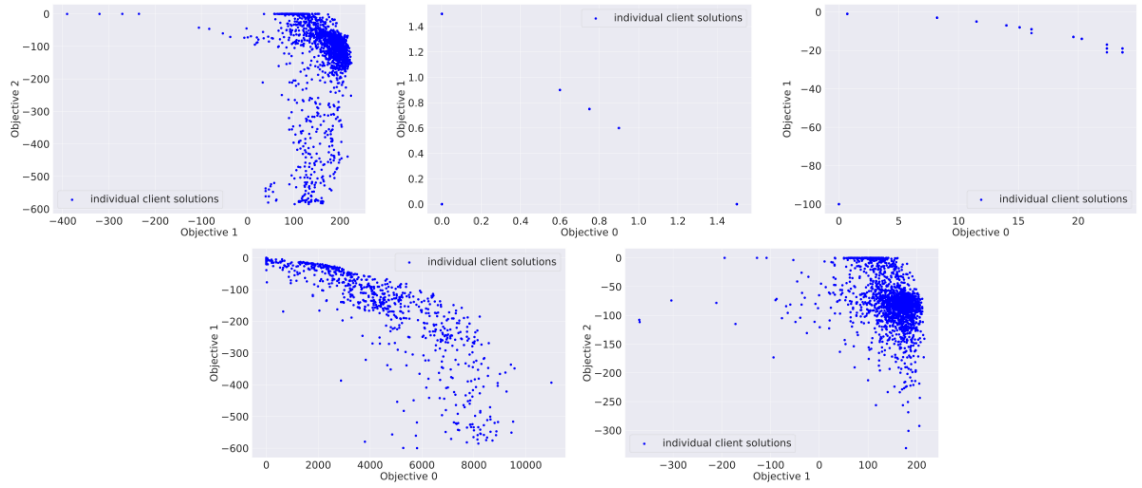


Figure 4.7: Sample illustrations of multi-objective solution spaces of different environments. Left to right: MO-Lunar Lander, Deterministic Minecart, Deep-Sea Treasure, MO-Halfcheetah and MO-LLcont. environments. For MO-LL, DMC, and MO-LLcont., results have dimension $4, 3$ and 4, respectively, and are here projected into a coordinate plane.

4.3.1 | Implementation and setup

We implement our experimental framework using the PyTorch package in the Python programming language. The code of our implementation is publicly available[1]. We evaluate our framework on a number of multi-objective reinforcement learning problems, as this class of problems represents a natural type of multi-objective problem well-suited to application and possesses a number of well-defined benchmarks [Fel23]. We run our experiments on five such MORL environments (illustrated in Figure 4.8): Deep-Sea Treasure [Vam11] (DST), Deterministic Minecart [Abe19] (DMC) and the multi-objective extension (MO-LL) of OpenAI's Lunar Lander gym environment, using a classical DQN algorithm [Mni15] to solve the scalarised RL problem on each client; and the multi-objective extensions of the halfcheetah (MO-HC) and the continuous variant of the Lunar Lander environment (MO-LLc.), using the DDPG algorithm. These five selected environments represent multi-objective problems with different characteristics: the Deep-Sea Treasure environment is relatively small and has a finite number of optimal solutions. The MO-Lunar Lander environment is more complex and has a large number of optimal or near-optimal solutions closely aligned in the solution space. Conversely, the Deterministic Minecart environment has a sparse reward space, leading to a very low number of optimal solutions, which are mutually distant in the solution space. The MO-Halfcheetah and Continuous Lunar Lander environments, finally, produce continuous rewards and so require a different RL algorithm to be solved, allowing us to examine the effectiveness of the FedPref algorithm across

1 https://gitlab.com/maria.hartmann/FedPref

different types of models. These differences, illustrated also in Fig. 4.7, present different challenges for federated aggregation.
The performance metric we report is the reward obtained by each client, according to its respective preference distribution. On all environments except MO-Halfcheetah, we run federated systems of 20 clients each; due to the higher complexity of the MO-Halfcheetah environment, we limit the number of clients to 10 in this case.

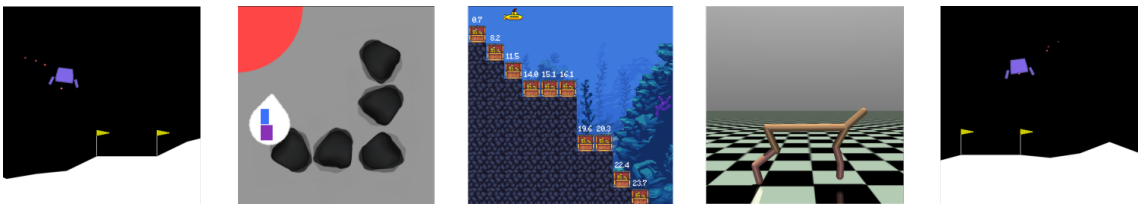


Figure 4.8: Illustration of multi-objective reinforcement learning environments used for validation experiments. Left to right: MO-Lunar Lander, Deterministic Minecart, Deep-Sea Treasure, MO-Halfcheetah and Continuous MO-Lunar Lander.

4.3.2 | Comparison to baselines

4.3.2.1 | Experiments

We compare our algorithm both to the classical baselines and to several algorithms developed to deal with other types of heterogeneity. As baselines, we run the same local learning algorithms with no communication between clients (no-communication) and the classical federated averaging (FedAvg) algorithm, aggregating all clients while disregarding heterogeneity. To the best of our knowledge, no previous algorithms that target this type of heterogeneity have been proposed in the literature; we therefore validate our approach against three additional algorithms from related fields that appear most relevant to our setting: FedProx [Li18], Many-Task Federated Learning [Cai23] (MaTFL) and Clustered Federated Learning [Sat19] (CFL). FedProx is a classical approach to the heterogeneity problem, commonly used as a baseline in data-heterogeneous settings. The underlying strategy appears intuitively to have the potential to transfer to the preference-heterogeneous setting, so we choose to retain this baseline. The MaTFL and CFL algorithms are chosen for the similarity of their approaches with the weighted aggregation and the clustering component of our algorithm, respectively; they also represent the two fields of Multi-Task Federated Learning and data-heterogeneous FL that we identified earlier in this chapter as most closely related to our problem setting. In addition to the standard versions of these algorithms from the literature, we also consider variants of the non-personalised algorithms that introduce a "fine-tuning" phase at the end of model training [Wan19]. Fine-tuning is a common strategy to allow a degree of personalisation between clients using an otherwise centralised training strategy. Following [Wan19], clients that perform fine-tuning end the training process with a single local training round instead of aggregating a global model. We label these variants as FedAvg+FT, FedProx+FT, CFL+FT, and FedPref+FT. We tune the hyperparameters for all algorithms via an initial grid search on a set of preferences sampled from a Dirichlet distribution. For each algorithm and environment, we select the best-performing hyperparameter configuration from this search. The details of this

parameter search and the local configurations of clients for each RL problem may be obtained from the supplementary material. Following the parameter tuning, we run all algorithms repeatedly, with client preference weights generated according to three different distributions: sampled from a Dirichlet distribution, sampled from a Gaussian distribution or weights generated to be equally spaced in the weight simplex. We report the results for all algorithms and distributions in Table 4.1.

4.3.2.2 | Analysis

We report the numerical results obtained for all algorithms and distributions in Table 4.1. In the remainder of this section, we discuss and contrast these results separately by preference distribution, from the most "extreme" preference differences between clients – the equidistant distribution – over the Dirichlet distribution to the Gaussian distribution, where client preferences are most similar.

Equidistantly distributed preference weights.
Under the equidistant distribution of preference weights across clients, we observe that variants of the FedPref algorithm outperform all other algorithms quite significantly on four out of five environments. On the MO-Halfcheetah environment, the FedPref+FT algorithm not only yields the highest average client reward of 3168.13, but it also has a notably lower variance than all other algorithms. Similarly, on the Continuous MO-Lunar Lander environment, FedPref-FT achieves a markedly higher average client reward score than all those compared. For the MO-LL environment, clients participating in the FedPref+FT algorithm obtain an average scalarised reward of 29.47, far ahead of the second-highest result of 18.49 achieved by another algorithm on the same environment. Indeed, the latter result is not accomplished by any federated algorithm, but by the baseline of non-communicating clients, with the remaining federated algorithms achieving much lower scores down to the lowest mean result of -94.06, returned by the CFL algorithm. Results for the Deep-Sea Treasure follow a similar pattern, while for the Deterministic Minecart environment no federated algorithm outperforms the result of the non-federated baseline. These results serve to underscore the difficulty of this heterogeneous distribution.

The case of equidistant preference weights likely represents the most "extreme" scenario among our experiments, where individual client objectives have on average the greatest mutual differences. In general, we would expect this to also map to greater differences in the models that match the preferences of each client, resulting in an advantage for those algorithms training personalised models. This is indeed illustrated in our results, reported in rows 11-20 of Table 4.1, as all variants of the FedProx and FedAvg algorithms perform notably worse in this scenario than for the other two types of preference distributions. This pattern persists across all experimental environments trained locally using the DQN algorithm (i.e. MO-LL, DMC and DST). More surprisingly, we also observe a poor performance by the CFL algorithm on these environments in this setting - further investigation, reported in the appendix, shows that CFL [Sat19] tends to yield highly unbalanced clusters in our experiments. As clusters in CFL train a single global model, this can lead to many clients with less personalised models, combined with a small number of clients that are separated early from the collaborative cluster – for this type of preference distribution, it is likely the large cluster that leads to a crucial lack of diversity. In other environments,

the clustering step of CFL is not triggered at all.

On the MO-Halfcheetah and Continuous MO-Lunar Lander environments, CFL and the two non-PFL algorithms perform notably better in comparison, though still worse than the FedPref algorithm. Indeed, in this scenario it becomes particularly important for any personalised algorithm to be able to accurately judge the compatibility of models, and to separate non-compatible models. This appears to be a strength of our algorithm: FedPref not only outperforms all others by a significant margin in four out of five environments. In the fifth environment - the Deterministic Minecart - none of the algorithms tested in our experiments perform better than the non-federated baseline. This might be the result of the high sparsity of the reward space, combined with the greater difference in client objectives, that make it difficult to group clients for aggregation.

With respect to the fine-tuning variants, we observe that the addition of a fine-tuning step does not generally lead to improved performances for the compared algorithms. A notable exception is the MO-LL environment, where the algorithms performing non-personalised aggregation deliver notably poor results without fine-tuning, with a drastic relative improvement with the addition of a fine-tuning step. However, the overall results in these cases are still markedly low. It appears that these algorithms fail to converge to a meaningful common solution across clients of such high preference diversity. In this context, disengaging from the federation naturally leads to improved local results.

Uniformly distributed preference weights.
In terms of expected client similarity, the Dirichlet preference distribution represents a "middle ground" between the other two types of distribution explored in this chapter. Preference weights are sampled uniformly at random from the weight simplex. Our results, reported in rows 1-10 of Table 4.1, show variants of the FedPref algorithm again outperforming all others on four out of five experimental environments. Compared to the results obtained under the equidistant preference distribution, some of the gains of the FedPref algorithm over those compared, though still existent, are less drastic, particularly on the relatively dense solution space of the MO-HC and MO-LL environments: on MO-LL, e.g. the FedProx algorithm yields a mean scalarised client reward of 31.2, relatively close to the top result of 37.27 achieved by the FedPref-FT algorithm. For the MO-LLc. environment, which appears to have an even higher localised density than MO-LL, the FedAvg algorithm even outperforms the FedPref algorithm under this distribution. However, the difference in favour of FedPref remains larger for the Deep-Sea Treasure environment, likely due to its discrete solution set: Here, the FedPref+FT variant of our algorithm obtains a mean scalarised client reward of 4.41, still followed by the no-communication baseline with an average reward of -0.43. The ranking of algorithmic results is similar on the Deterministic Minecart environment, though less decisive. It appears that the lower density of (optimal) solutions available in the latter two environments, combined with the intermediate objective heterogeneity of this setting, continues to present a difficult challenge to the federated algorithms from the literature.

The addition of a fine-tuning step again yields lower mean client scores for all algorithms performing non-personalised aggregation. In most cases, the results for the fine-tuned variant of an algorithm are markedly worse than for the same algorithm without fine-tuning. We theorise that over the course of the federated training process, clients in the federated

system have learned to converge to a mutually beneficial, globally optimal "compromise" solution. This compromise, however, is quite fragile, as all individual clients operate under different preferences, i.e. different loss functions, and the distance between client-optimal solutions and the global model appears too great to overcome during fine-tuning. The difference in performance between fine-tuning and non-fine-tuning variants is less great for the FedPref algorithm, most likely because diverse clients are separated more successfully at an earlier stage of the training process.

Gaussian-distributed preference weights.
Finally, in the setting where weight preferences are drawn from a Gaussian distribution, three different algorithms achieve the top scores for different environments (see results in row 21-30 of Table 4.1): Results on the Deep-Sea Treasure environment remain dominated by both variants of the FedPref algorithm, with no other federated algorithm outperforming the non-federated baseline. Similarly, FedPref-FT outperforms all others on the MO-LL and MO-Halfcheetah environments. For the compared algorithms, the addition of a fine-tuning step to the various algorithms has a similarly negative effect as in the experiments under a Dirichlet preference distribution.

Under the Gaussian distribution, clients are more likely to have more similar preferences, potentially supporting more similar models. In this case, plain (equally-weighted) aggregation appears to do well, with the CFL algorithm delivering the second-best performance on the MO-LL environment. The two non-PFL algorithms also perform notably better under this preference distribution than in the other two settings - in fact, in this case the plain FedAvg algorithm outperforms all others in the DMC environment, and the FedProx algorithm achieves the top result in the Continuous MO-Lunar Lander environment. The former result may in part be owing to the sparse solution space of the problem, with the clients in federation jointly converging on a single local optimum; but it is nonetheless part of a wider trend. In contrast to the very different performance of the compared algorithms on some environments, FedPref appears to adapt quite well to this setting, delivering the best performance in two environments and the second-best in two others.

In general, these results indicate that the FedPref algorithm is capable of adapting to a range of different preference distributions and problem types, outperforming all compared algorithms in the majority of experiments. In almost all cases where our algorithm does not deliver the best performance, it is outperformed by only one other, and by different algorithms for different problems. Furthermore, these results are preserved across different local training algorithms and different model architectures. This shows the high flexibility and robustness of our algorithm, making it a good overall choice in the general case, where the distribution of preference weights of the characteristics of the learning problem may be unknown.

4.3.3 | Ablation study

We perform an ablation study of our algorithm, comparing the performance of the full algorithm with that of its individual components, i.e. performing only the weighted aggregation strategy or only the clustering strategy, respectively. For all three configurations, we also consider variants that perform a single round of fine-tuning at the end of the training process. The hyperparameters of all versions remain fixed to the values obtained for our

Table 4.1: Experimental results comparing our proposed FedPref algorithm to MaTFL, CFL, FedProx, FedAvg and individual learning without cooperation.

		MO-LL(↑)	DMC(↑)	DST(↑)	MO-HC(↑)	MO-LLc.(↑)
Dirichlet	No comm.	14.32 σ13.3	−2.52 σ0.9	−0.43 σ1.8	2440.30 σ511.3	11.95 σ12.6
	FedAvg	30.52 σ14.7	−3.20 σ3.4	−16.50 σ24.7	2234.06 σ1202.9	**34.19** σ11.4
	FedAvg+FT	4.31 σ11.0	−5.41 σ0.9	−36.40 σ10.0	2901.84 σ798.1	16.29 σ6.3
	FedProx	31.20 σ16.9	−3.34 σ3.6	−11.05 σ22.1	2172.35 σ1231.2	32.45 σ11.2
	FedProx+FT	3.21 σ11.3	−5.44 σ0.6	−34.52 σ7.2	3017.22 σ795.3	13.82 σ12.5
	CFL	31.18 σ17.8	−2.76 σ0.9	−12.90 σ21.8	2835.50 σ817.7	26.90 σ12.5
	CFL+FT	10.09 σ13.2	−3.82 σ2.1	−35.14 σ10.2	2864.62 σ871.0	19.81 σ7.6
	MaTFL	7.82 σ9.9	−4.39 σ2.2	−6.32 σ3.6	1596.59 σ558.9	10.98 σ7.3
	FedPref+FT	32.22 σ11.3	−2.42 σ1.6	**4.41** σ1.7	2980.26 σ784.4	32.17 σ7.0
	FedPref-FT	**37.27** σ11.9	**-1.90** σ1.2	1.21 σ3.4	**3104.59** σ742.3	32.91 σ9.8
Equidist.	No comm.	18.49 σ10.5	**-1.70** σ1.5	0.68 σ1.9	2265.67 σ440.5	22.49 σ6.9
	FedAvg	−55.64 σ46.9	−6.77 σ0.3	−17.87 σ26.2	1952.42 σ321.4	28.04 σ6.8
	FedAvg+FT	−16.95 σ5.6	−6.24 σ0.4	−36.01 σ3.1	3023.42 σ204.2	6.42 σ8.1
	FedProx	−73.98 σ45.3	−6.75 σ0.2	−23.23 σ26.8	1948.46 σ532.2	27.75 σ11.5
	FedProx+FT	−9.69 σ4.9	−6.08 σ0.4	−36.17 σ2.6	3019.46 σ206.9	1.01 σ8.2
	CFL	−94.06 σ30.8	−2.74 σ0.5	−34.62 σ23.5	2831.28 σ304.4	25.97 σ4.6
	CFL+FT	4.70 σ5.6	−2.35 σ0.7	−37.17 σ2.9	3097.66 σ282.7	11.71 σ10.4
	MaTFL	11.99 σ6.5	−4.03 σ1.2	−6.16 σ2.8	1643.90 σ239.3	19.20 σ5.6
	FedPref+FT	**29.47** σ3.5	−2.21 σ1.7	**2.78** σ2.5	**3168.13** σ145.7	36.52 σ4.8
	FedPref-FT	29.26 σ4.2	−2.35 σ1.8	2.26 σ2.2	3044.61 σ239.9	**45.28** σ11.8
Gaussian	No comm.	15.33 σ13.4	−3.61 σ2.2	1.13 σ0.9	2575.39 σ790.7	17.08 σ5.9
	FedAvg	31.53 σ13.1	**-2.47** σ3.3	−13.32 σ25.6	2028.17 σ1282.5	33.15 σ11.3
	FedAvg+FT	14.99 σ9.7	−3.87 σ2.1	−37.42 σ5.5	2758.24 σ1081.8	19.37 σ8.5
	FedProx	32.75 σ13.0	−2.94 σ3.1	−2.07 σ16.0	2042.50 σ1183.2	**33.31** σ7.3
	FedProx+FT	14.61 σ10.1	−4.41 σ1.7	−37.15 σ9.1	2732.12 σ940.0	19.72 σ8.0
	CFL	37.73 σ11.5	−4.00 σ1.8	−24.82 σ27.5	2635.25 σ812.3	25.56 σ8.0
	CFL+FT	18.30 σ9.2	−3.71 σ1.6	−33.75 σ8.3	2852.83 σ776.1	16.41 σ8.5
	MaTFL	6.48 σ9.4	−5.04 σ1.1	−6.24 σ3.4	1355.33 σ489.7	13.36 σ7.1
	FedPref+FT	36.43 σ7.3	−2.53 σ1.8	2.87 σ2.6	2786.28 σ888.8	33.06 σ6.7
	FedPref-FT	**38.90** σ8.8	−2.86 σ1.5	**2.88** σ3.4	**2923.27** σ897.0	33.18 σ7.7

algorithm in previous experiments. The results, reported in Table 4.2, show different outcomes for the different types of problems we study. The effect of fine-tuning, too, appears to vary for different environments.

For the Deterministic Minecart environment with its very sparse reward space, we observe that the Clustering+FT component performs better individually than combined – the Clustering+FT component achieves the highest average scalarised client reward of −1.53, whereas the combined components yield a mean reward of −1.90 without fine-tuning. In this case, it is likely that the individual clients' preferences ultimately lead to very different optimal models, with less benefit obtained from extensive cooperation between different models. Separating clients early enough during the training process would then be crucial, before a "compromise" model emerges that differs greatly from individually optimal models. Otherwise, such a consensus model might be so different from the optimal fit for an individual clients' preferences that the latter becomes too hard to recover in

training even if clients are separated at a later stage.
As a side remark, we note that this hypothesis is also supported when comparing the results for the FedProx and FedAvg algorithm, discussed in Section 4.3.2, Table 4.1, for this environment. The approach behind these two algorithms forces a high level of collaboration between the clients, and does not lead to high overall results for this environment when preferences are sufficiently different. Even fine-tuning does not generally deliver improvements, likely because the at the end of training, the global consensus model is too far distant from locally optimal models to reach.
Returning to the ablation study, we suggest that the clustering component without weighted aggregation likely succeeds more quickly in separating very different models early during the training process, with the cluster-mean model converging more definitively. This separation appears not fully effective, as seen by the poor performance of the clustering component without fine-tuning; yet it succeeds in enabling the training of models that are sufficiently diverse that a single fine-tuning round can ameliorate these problems. The weighted-aggregation component in isolation and the full FedPref algorithm appear slightly less successful at separating diverse clients early during training, leading to consensus models that do not improve with fine-tuning. In spite of this, variants of all three versions succeed in outperforming the compared approaches under the Dirichlet distribution – see Table 4.1.
Interestingly, the success of the individual clustering component over the FedPref algorithm is reversed on DST, the other environment with a sparser solution space. Here, the combination of clustering and weighted aggregation appears to encourage effective separation. In contrast, we observe significantly improved results for the combination of both components over each component individually for the MO-LL and MO-HC environments, both when comparing fine-tuning variants and between non-fine-tuning variants. For the MO-Halfcheetah environment, we observe rather similar average client performances for variants of all three configurations, with the FedPref-FT algorithm outperforming the other configurations. Once again, fine-tuning slightly decreases performance for the clustering component and the combined components, indicating that perhaps in these cases some clients might have benefited from earlier separation. However, this effect is not strong here. The results for the Continuous MO-Lunar Lander show a quite similar pattern to the MO-LL environment, with the main difference being the relatively weaker performance of FedPref-FT. In this case, the Clustering-FL component outperforms both FedPref variants.
We draw three general impressions from the ablation study:

- A variant of the FedPref algorithm performs better than its individual components on three out of five environments, and results near to the highest score in the other two cases.
- The success of each component appears related to the characteristics of the problem being solved.
- The effect of a fine-tuning step may serve as a useful indicator of the effectiveness of diverse clients' cooperation within the FedPref algorithm.

Table 4.2: Experimental results comparing the mean reward achieved by the individual components of our algorithm.

	MO-LL	DMC	DST	MO-HC	MO-LLc.
Clustering only + FT	27.27 σ13.9	**-1.53** σ0.8	0.95 σ3.8	2940.45 σ982.4	28.24 σ9.7
Clustering only - FT	34.21 σ16.7	−4.27 σ7.1	0.55 σ3.1	3049.63 σ683.1	**34.58** σ15.4
Weighted agg. only + FT	17.70 σ9.8	−2.20 σ0.9	−31.24 σ8.5	3044.93 σ821.6	26.12 σ9.3
Weighted agg. only - FT	32.54 σ21.1	−2.08 σ1.1	−21.83 σ13.6	2425.13 σ1490.2	34.36 σ15.1
FedPref+FT (combined)	32.22 σ12.0	−2.42 σ1.7	**4.41** σ1.8	2980.26 σ826.8	32.17 σ7.4
FedPref-FT (combined)	**37.27** σ12.5	−1.90 σ1.2	1.21 σ3.6	**3104.59** σ782.5	32.91 σ10.4

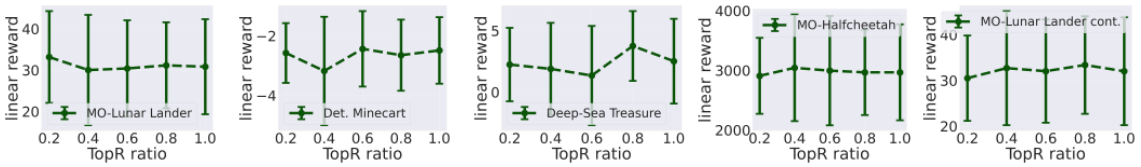


Figure 4.9: Impact of the choice of *topR* parameter on average reward obtained by clients. Left to right: results for MO-Lunar Lander, Deterministic Minecart, Deep-Sea Treasure, MO-Halfcheetah and Continuous MO-Lunar Lander environments.

4.3.4 | Impact of topR parameter and similarity bound

We study the performance impact of the choice of two hyperparameters that are integral to our algorithm: the parameter R for the *topR* operator, and the lower similarity bound used in computing aggregation weights. The parameter R describes the proportion of each model layer to be used by our metric in calculating similarity (see Equations 4.4 and 4.3, respectively, for the definitions of the *topR* operator and our similarity metric).

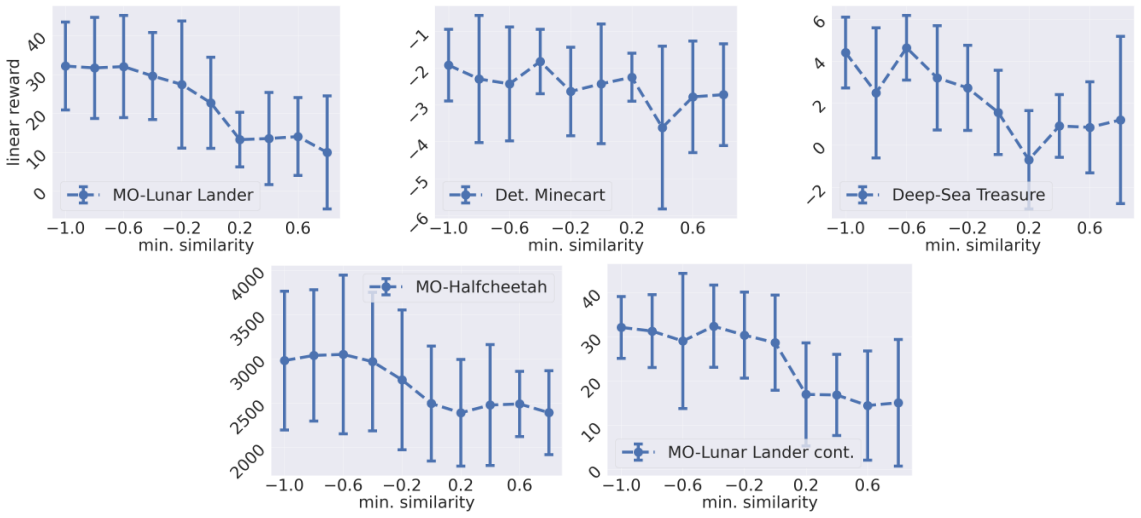


Figure 4.10: Impact of the choice of minimum-similarity threshold on average reward obtained by clients. Left to right: results for MO-Lunar Lander, Deterministic Minecart, Deep-Sea Treasure, MO-Halfcheetah and Continuous MO-Lunar Lander environments.

For an intuition on the meaning of R, consider that R describes the proportion of model parameters to be compared when rating the similarity of two clients. The higher R is, the more parameters are taken into account; for $R = 1$, all model parameters are compared, recovering the "normal" cosine similarity metric. The minimum similarity bound is used during the weighted aggregation step to include only models exceeding a given similarity value in the aggregation.

The results nevertheless support the use of this modified similarity metric: the use of a well-tuned *topR* parameter is shown to improve performance compared to the standard metric that is recovered with $R = 0$ in four out of five studied environments, in some cases quite significantly. For the MO-LL environment, the highest mean scalarised client reward of 33.18 is obtained for $R = 0.2$, representing an improvement of approximately 7.6% over the result of 30.84 for $R = 0$; the most successful configuration on the MO-LLc. environment leads to a circa 4.2% higher mean reward. For the DST environment, the improvement is even greater: from 2.52 for $R = 0$ to 3.76 for $R = 0.8$, an increase of roughly 49%.

The relative improvement for the MO-HC environment is somewhat lower, but it does exist: a *topR* parameter of 0.4 shows roughly a 2.6% improvement over the plain metric, from 2967.83 to 3043.55. From these observations, we conclude that a modification of the plain cosine similarity metric for quantifying model similarity does have promise; however, the high variance we observe during the sensitivity across all environments indicates that the stability of such a metric leaves room for improvement.

The results for the minimum-similarity threshold (see Fig. 4.10) show commonalities across all five environments, suggesting that thresholds lower than 0 are remarkably beneficial to the learning outcome of our algorithm: the relative improvement in mean scalarised client reward between a similarity threshold of 0 and the optimal discovered value ranges from 13% for the MO-LLc. environment with threshold -0.4 to a full 296.8% improvement for the DST environment with threshold -0.6. Indeed, it appears that this pattern is in general quite stable, so fixing the minimum-similarity threshold to -1 even without tuning this parameter is likely to lead to good results.

Though counter-intuitive at first glance, given the geometric interpretation of cosine similarity, this outcome is quite reasonable in the context of our algorithm. Firstly, we note that the purpose of our algorithm's clustering strategy is to group those nodes into clusters that can benefit from collaboration. Hence, improved results for a lower minimum-similarity threshold indicate that this grouping is successful, as even relatively dissimilar clients inside the same cluster improve with collaboration. Secondly, the fact that clients train personalised and therefore different models means that some dissimilarity is induced by definition of the metric, through the choice of the cluster-mean model as a reference point in computing the cosine similarity.

4.3.5 | Validation of clustering strategy

We validate the clustering strategy on all three environments by running FedPref on several artificially constructed configurations where multiple clients share the same preferences. We construct two types of configurations: one where preferences are distributed among equal numbers of clients each (4 distinct preference weights, with each preference weight held by 5 clients), and one where the number of clients varies for each preference (4 distinct

preference weights, held by $2, 3, 6$ and 9 clients, respectively). We observe how well the clustering algorithm groups similar clients, and we study how the similarity between clients develops during training. Due to scope constraints, we present only one such configuration here; visualisations of other configurations are included in the supplementary material.
Fig. 4.11 shows client similarity values at selected steps of the training process on the MO-Lunar Lander environment under the balanced preference distribution. (Note that, for ease of visualisation, clients with the same preference weight are grouped together by index.) The evolution of client clusters over the duration of the training process is visualised in Fig. 4.12. In this illustration, clients with the same preferences are represented as boxes of the same colour; boxes that form a connected bar represent a cluster. Note that the relative positioning of boxes does not necessarily correspond to client index. We see in the visualisation that client models initially develop individually, but varying preference similarity is already reflected in the model similarity computed by our metric. At the earliest visualised stage (after five aggregation steps; left-most image in Fig. 4.11), all clients are still grouped together in a single cluster. Nevertheless, the weighted aggregation strategy gives individual models the freedom to develop separately, yet also appears to be successful in encouraging the aggregation of clients with the same objectives.
In the second image, at an intermediate stage of the training process, a split into multiple clusters has occurred. The grouping of clients with the same preferences is preserved across experimental runs, but multiple groups of such clients continue to collaborate at this training stage, with different groups clustered together in different experimental runs. In the instance visualised here, clients $1-5$ and $11-15$ are all contained in the same cluster.
In the final image, close to the end of the training phase, we observe that the similarity of the models obtained by clients with the same preferences is very high, while the similarity to other models appears lower than before. This indicates that these clients have been separated into individual clusters, and that the personalised models within these clusters are converging; Fig. 4.12 confirms this impression. In this visualisation of clustering states, we see that all sets of clients with the same preferences have been separated correctly by the end of aggregation round 16.

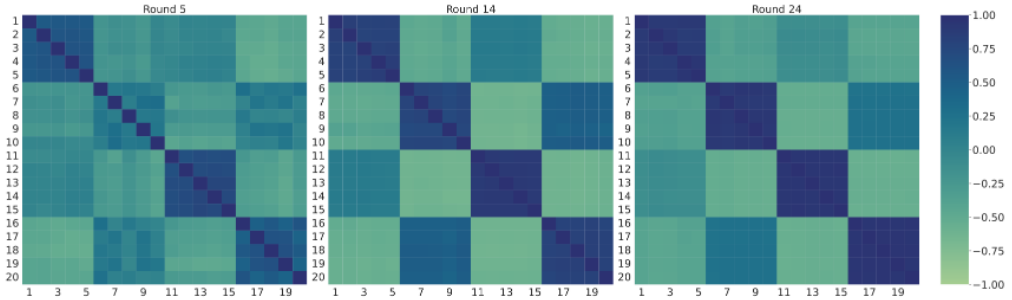


Figure 4.11: Mutual client similarity at different stages during a single experimental run on the MO-LL environment. Left to right: client similarities after aggregation round $5, 14$ and 26 of 28, respectively.

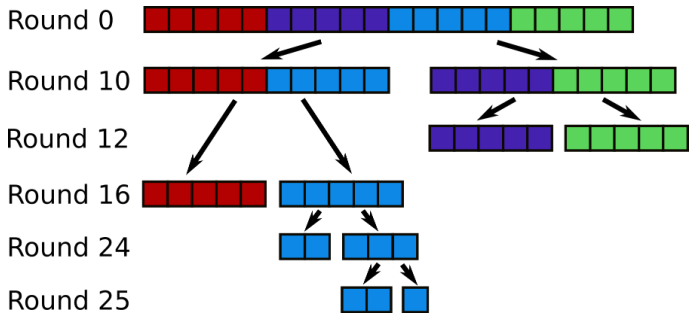


Figure 4.12: Cluster states at different training stages during a single experimental run on the MO-LL environment. Clients with the same preferences are represented as boxes of the same colour.

4.4 | A different point of view: multi-objective evaluation

Up to this point, we have considered the preference heterogeneity problem only from the traditional client-level FL viewpoint, with the sole aim of optimising the performance of each client according to its preferences. However, we note that a system-level view is also particularly relevant in this preference-heterogeneous setting, and should be considered in judging the performance of any algorithm designed to solve it. Recall that in this setting, clients are solving a problem with multiple objectives, with each client's preferences describing the relative importance of each objective. The general aim of modelling a problem with multiple objectives is to capture the inherent complexity of the real world, allowing for the consideration of different, potentially conflicting influences. In solving the problem for different preference weights, the assumption is that these preferences will be met in a meaningful way, i.e. that different preference distributions will in fact lead to substantially different trade-off solutions. Any algorithmic approach that fails to do so arguably largely invalidates the premise for using multiple objectives in the first place. Our algorithm endeavours to meet this underlying expectation, allowing client models to diverge even while aggregating related models.

To evaluate this aspect of algorithmic performance, we propose to use standard multi-objective metrics from the field of multi-objective optimisation (MOO) to evaluate the set of solutions generated by all federated clients under the different preference distributions. In this chapter, we focus on four common state-of-the-art metrics. Three of these metrics were previously introduces in Chapter 3: the hypervolume [Zit99], sparsity [Xu20] and inverted generational distance (IGD) [Coe05]. In addition to these metrics, we also report the cardinality of solutions in this chapter. Cardinality is the number of solutions in a given set that lie on the Pareto front.Recall that these metrics quantify different properties of the solution set: sparsity measures the diversity of the solution set, IGD the convergence, and the hypervolume metric a combination of both. A desirable set of solutions would have low sparsity and IGD values and a high hypervolume.

In the remainder of this section, we demonstrate the evaluation of our experiments using these metrics and discuss the results.

4.4.1 | Experimental evaluation using multi-objective metrics

We re-evaluate the same experiments presented and discussed in Section 4.3 under multi-objective aspects. To the best of our knowledge, the performance of federated algorithms under multi-objective aspects has not been discussed before; hence there are no obvious additional baselines to consider for comparison. We note that in this setting, a main challenge for federated system can be expected to lie in the need for solution diversity, as client models trained in federation need to achieve some level of convergence to effectively exchange information. Indeed, the classical FL baseline of a system of clients without communication continues to provide a relevant challenge here, as non-cooperating clients might be expected to achieve a high diversity of solutions by default. The four multi-objective metrics are reported in Table 4.3 (Hypervolume), Table 4.5 (IGD), Table 4.4 (sparsity) and Table 4.6 (cardinality).

Equidistantly distributed preference weights. We first observe that a variant of the FedPref algorithm generates or matches the highest cardinality, i.e. the number of solution on the Pareto front, in three out of five experimental environments. For example, in the Deep-Sea Treasure environment, the mean cardinality achieved by FedPref is 8.0, meaning that of the 20 clients per federated system, on average 8 clients find a distinct optimal trade-off solution. This stands in marked contrast to the non-PFL algorithms and CFL, which find only 1 and 1.2 such solutions, respectively, or ≤ 4.0 and 2.2 for the corresponding fine-tuned variants. These algorithms achieve higher cardinalities in environments with more dense solution spaces, most notably MO-LL and MO-LLc. However, these higher cardinalities are likely caused mainly by statistical differences in model evaluations. The fine-tuning variants of these algorithms also yield higher cardinality scores in many cases, as might be expected; yet these remain generally lower than the highest scores. In the Deterministic Minecart environment, the mean cardinality of 3.2 obtained by the FedPref+FT algorithm is only beaten by the value of 3.3 of the CFL+FT algorithm and the non-federated baseline, and only by a small margin. These results match our observations of the success of each algorithm when considering average scalarised client performance in Section 4.3.

The corresponding hypervolume results support the overall impression given by the examination of cardinality values: higher cardinality values correspond to the higher hypervolume values achieved for each experimental environment, though the overall ranking does not translate exactly. For example, in the MO-Halfcheetah environment, where the FedPref-FT and FedProx+FT algorithms obtain cardinality values of 6.9 and 7.0, respectively, the FedPref-FT algorithm nevertheless yields a higher average hypervolume, indicating that a solution set with higher diversity or convergence was found. A comparison of sparsity and IGD results suggests the former explanation.
Another notable exception to this observed correlation between high cardinality and high hypervolume occurs when comparing algorithms with their fine-tuning variants. Although the fine-tuning step generally appears to yield more distinct optimal solutions, this does not always translate to equally great improvements in hypervolume. This is likely because new solutions obtained by fine-tuning do not diverge too far from others.

Uniformly distributed preference weights. On average, more than half of all

federated clients solving the MO-LL, MO-HC and MO-LLc. environments with a variant of the FedPref algorithm find a distinct optimal trade-off solution in all experimental configurations. For the DST environment, which has only 10 discrete solutions in total, the FedPref+FT algorithm consistently leads to the identification of more than 70% of possible solutions. As for the previous distribution, a variant of FedPref achieves the highest cardinality for the MO-LL, DST, and MO-LLc. environments, including the highest overall cardinality of 13.7 for the MO-Lunar Lander. On the other two environments, the number of trade-off solutions found for FedPref is again only slightly lower than the highest achieved by any algorithm. The effects of adding a fine-tuning step to the various algorithms appear similar to the those discussed for the equidistant preference distribution. Similarly to previous observations, the number of optimal trade-off solutions found appears to translate well to higher observed hypervolume values, with a variant of FedPref accomplishing the highest rank in the same three environments as for the cardinality. This indicates that the set of solutions found by the federated system executing FedPref does have a high level of diversity, and that the comparably high sparsity values we observe for these same configurations are likely due to greater spread of optimal solutions. As a final observation, we note that in many cases, the FedPref algorithm yields more distinct optimal trade-off solutions than the non-federated baseline, suggesting that PFL can assist in effectively exploring the solution space and finding an even more diverse solution set than non-collaborative clients.

Gaussian-distributed preference weights. For this distribution, finding a diverse set of solutions appears to present a particular challenge, probably because preferences are more likely to be more similar here, allowing clients to jointly exploit local optima more successfully. With respect to the cardinality metric, we observe a similar pattern as before, with a variant of the FedPref algorithm again obtaining or matching the highest cardinality values in three out of five environments, and nearing the highest value in all others. However, the hypervolume results are less decisive here than for the other two distributions: FedPref does reach the highest hypervolume value on only three out of five environments. However, on the MO-HC environment, the highest hypervolume score is reached by the non-federated baseline, with FedPref following in second place, and this ranking is only just reversed on the DST environment. It is likely that in this preference distribution setting, the federated clients succeed more readily in collaborating, leading to higher individual results, as seen in the analysis in Section4.3.2. The down-side of this enhanced collaboration could be a loss of diversity, as indicated by the slightly higher hypervolume values accomplished by the non-federated baseline, where clients do not collaborate at all, in this case. Nonetheless, consideration of the corresponding sparsity and IGD values shows that variants of FedPref yield lower results than the non-federated baseline for both metrics. This indicates that the set of results found by FedPref is overall more evenly distributed.

In conclusion, in analysing multi-objective metrics across all five environments, we observe that the FedPref algorithm leads most consistently of all federated algorithms to a diverse set of good trade-off solutions. A general challenge from a multi-objective viewpoint is the lack of solution diversity brought on by the aggregation of client models. This is most evident in the results of the FedAvg and FedProx algorithms, which find generally low

numbers of distinct optimal trade-off solutions, even with the addition of a fine-tuning phase. Despite the relatively high mean scalarised reward we have observed in the client-level evaluation of the previous section, these results are arguably not very satisfactory from a multi-objective point of view of the system. The compared personalised FL algorithms, CFL and MaTFL, generally perform somewhat better, with variants of CFL in particular achieving a relatively high solution diversity at times. This behaviour may stem from the unbalanced clustering strategy of CFL, which we have previously remarked upon in Section 4.3.2. However, unlike for FedPref, the performance of CFL is not consistent across preference distributions and environments. As with the client-level evaluation, FedPref proves to be the most adaptable federated algorithm of all those evaluated, both with respect to different types of preference heterogeneity and multi-objective problems with different characteristics.

4.4.2 | Ablation study - multi-objective performance

In this section, we briefly revisit our previously discussed ablation study under multi-objective aspects. The corresponding multi-objective metric results are shown in Table 4.7. For four out of five experimental environments, the results quite clearly indicate that a variant of the FedPref algorithm yields a better performance under multi-objective aspects than either of its components in isolation. For the MO-LLc. environment, the optimal values for three out of four metrics are obtained by the FedPref algorithm. For each of the MO-LL, DMC, and MO-LLc. environments, a variant of the combined algorithm achieves or matches both the highest hypervolume and cardinality, and in both cases also the lowest IGD value. In combination, these results indicate that the solution sets obtained by the respective federated systems do accomplish the highest diversity and convergence, and that the elevated sparsity in these cases is simply a consequence of the overall wider distribution of solutions. Results for the DST algorithm give a similar impression.

The results for the MO-HC environment follow a slightly different pattern: FedPref-FT yields the highest hypervolume value in this case, but the best scores for the remaining metrics are obtained by variants of the Weighted aggregation component. It appears that this component in isolation achieves a higher convergence than the FedPref algorithm, at the cost of a reduction in the diversity of solutions.
Finally, we note that the addition of a fine-tuning step at the end of the local training phase generally does not greatly impact the multi-objective results in most cases. While variants with fine-tuning tend to produce higher cardinalities (i.e. more optimal trade-off solutions), this does not generally translate to improved results in the other metrics. A brief fine-tuning phase is likely not sufficient to significantly increase the diversity of the solution set; new trade-off solutions discovered during fine-tuning would tend to be very similar to each other.

Table 4.3: Hypervolume(↑) metric for multi-objective solutions obtained by our proposed FedPref algorithm, compared to MaTFL, CFL, FedProx, FedAvg and individual learning without cooperation. Where indicated in the header, both the main value and the standard deviation value have been divided by the given power.

		MO-LL ($\cdot 10^7$)	DMC	DST ($\cdot 10^1$)	MO-HC ($\cdot 10^4$)	MO-LLc. ($\cdot 10^7$)
Dirichlet	No comm.	175.54 σ7.1	198.01 σ28.2	204.47 σ20.1	429.78 σ45.7	77.20 σ10.9
	FedAvg	176.53 σ7.6	90.12 σ60.3	88.89 σ72.6	242.72 σ67.5	91.20 σ5.1
	FedAvg+FT	169.13 σ20.8	142.99 σ13.9	134.48 σ53.6	410.93 σ23.2	94.55 σ9.7
	FedProx	172.67 σ13.5	90.13 σ60.3	103.10 σ67.5	235.33 σ65.8	90.11 σ6.4
	FedProx+FT	169.05 σ24.5	151.93 σ18.1	147.67 σ53.5	419.97 σ19.0	92.10 σ7.8
	CFL	178.41 σ14.8	**209.29** σ15.7	131.44 σ74.3	406.33 σ37.1	94.76 σ9.7
	CFL+FT	173.04 σ14.4	206.50 σ24.3	172.20 σ26.7	395.06 σ39.6	92.26 σ7.8
	MaTFL	163.81 σ14.6	86.38 σ36.7	214.25 σ18.3	299.03 σ49.5	80.16 σ7.8
	FedPref+FT	193.94 σ7.5	202.79 σ29.7	**220.36** σ10.3	421.43 σ39.5	100.04 σ5.5
	FedPref-FT	**201.97** σ7.0	203.74 σ13.2	218.04 σ18.3	**435.87** σ36.5	**101.57** σ8.8
Equidist.	No comm.	156.00 σ25.3	**204.43** σ13.2	216.56 σ13.8	424.79 σ53.5	53.73 σ13.0
	FedAvg	69.10 σ48.6	29.80 σ0.2	89.88 σ73.3	235.81 σ31.0	78.96 σ9.6
	FedAvg+FT	85.78 σ21.2	138.32 σ40.9	157.36 σ3.6	426.24 σ25.8	54.83 σ13.5
	FedProx	46.44 σ38.5	29.85 σ0.1	74.04 σ74.0	247.38 σ32.7	77.25 σ10.9
	FedProx+FT	96.54 σ13.4	137.51 σ15.4	161.75 σ23.9	426.88 σ31.8	50.86 σ14.5
	CFL	29.76 σ28.3	183.64 σ33.4	45.29 σ69.0	414.97 σ31.5	80.58 σ17.8
	CFL+FT	130.92 σ11.9	202.94 σ14.0	171.12 σ24.4	427.94 σ39.4	65.00 σ16.8
	MaTFL	150.15 σ25.9	79.08 σ36.0	200.99 σ25.8	290.91 σ41.9	58.93 σ12.8
	FedPref+FT	166.37 σ19.1	197.45 σ27.6	222.37 σ8.5	**442.63** σ18.4	74.28 σ5.8
	FedPref-FT	**182.90** σ11.4	199.28 σ8.4	**225.23** σ4.9	441.17 σ27.3	**101.04** σ6.7
Gaussian	No comm.	176.61 σ11.7	201.23 σ11.8	221.37 σ8.1	**453.61** σ43.9	83.10 σ7.6
	FedAvg	178.50 σ8.2	92.04 σ58.6	99.25 σ65.4	220.48 σ95.6	90.14 σ6.3
	FedAvg+FT	177.79 σ13.2	126.71 σ43.0	149.97 σ58.5	402.24 σ39.8	98.86 σ4.4
	FedProx	181.00 σ10.7	69.89 σ53.2	133.89 σ44.7	222.09 σ83.4	89.06 σ6.6
	FedProx+FT	174.14 σ11.5	133.09 σ31.0	143.09 σ15.2	398.41 σ41.5	96.28 σ4.8
	CFL	187.52 σ6.2	204.03 σ23.4	97.56 σ82.0	413.57 σ28.5	91.74 σ5.2
	CFL+FT	181.28 σ7.8	**215.13** σ13.3	159.97 σ19.9	437.62 σ27.5	92.08 σ10.9
	MaTFL	167.94 σ16.8	101.00 σ46.9	201.87 σ27.8	280.87 σ45.7	76.94 σ12.2
	FedPref+FT	197.28 σ8.4	195.67 σ38.2	215.89 σ10.1	420.45 σ47.7	**101.23** σ7.9
	FedPref-FT	**202.06** σ4.9	203.88 σ13.4	**221.44** σ10.7	450.20 σ25.0	100.21 σ7.0

Table 4.4: Sparsity(↓) metric for multi-objective solutions obtained by our proposed FedPref algorithm, compared to MaTFL, CFL, FedProx, FedAvg and individual learning without cooperation. Lower sparsity means that the mutual distance between solutions obtained by the algorithm is lower. The metric has zero-value by definition if there is only one solution on the Pareto front.

		MO-LL ($\cdot 10^2$)	DMC	DST	MO-HC ($\cdot 10^4$)	MO-LLc. ($\cdot 10^1$)
Dirichlet	No comm.	68.97 σ94.1	2.00 σ0.8	27.60 σ8.6	488.85 σ318.4	246.92 σ95.3
	FedAvg	3.29 σ2.5	**0.00** σ0.0	**0.00** σ0.0	7.75 σ18.7	**27.71** σ20.0
	FedAvg+FT	64.07 σ60.6	0.01 σ0.0	9.42 σ6.5	296.29 σ104.7	477.85 σ302.2
	FedProx	2.43 σ1.6	0.00 σ0.0	0.00 σ0.0	**2.62** σ5.1	31.37 σ27.2
	FedProx+FT	59.04 σ47.5	0.23 σ0.4	16.70 σ16.0	278.82 σ134.1	338.70 σ269.5
	CFL	**1.47** σ1.0	1.03 σ0.5	34.29 σ72.5	341.84 σ208.5	221.40 σ93.5
	CFL+FT	85.21 σ135.7	0.94 σ0.2	42.52 σ71.0	333.20 σ145.3	334.66 σ298.9
	MaTFL	25.65 σ13.2	1.46 σ1.5	57.06 σ29.0	380.86 σ398.7	247.25 σ194.1
	FedPref+FT	19.35 σ8.6	1.83 σ1.0	28.00 σ7.6	408.45 σ167.6	194.61 σ88.7
	FedPref-FT	19.38 σ17.0	1.88 σ0.8	33.07 σ7.9	517.10 σ377.2	92.45 σ68.8
Equidist.	No comm.	200.89 σ307.6	2.63 σ2.1	25.12 σ5.2	301.92 σ126.3	513.09 σ189.3
	FedAvg	8.74 σ3.5	**0.00** σ0.0	**0.00** σ0.0	**3.43** σ7.9	31.79 σ9.5
	FedAvg+FT	384.42 σ162.3	0.47 σ1.3	10.25 σ3.0	250.54 σ79.1	991.11 σ659.6
	FedProx	5.31 σ1.9	0.00 σ0.0	0.00 σ0.0	33.23 σ60.2	**24.54** σ25.5
	FedProx+FT	278.97 σ159.3	0.02 σ0.1	17.59 σ17.6	288.65 σ144.8	814.88 σ382.5
	CFL	**4.29** σ2.6	1.66 σ0.8	0.44 σ0.9	338.76 σ139.4	512.27 σ391.4
	CFL+FT	626.89 σ420.1	1.77 σ0.7	20.79 σ37.7	288.57 σ99.2	606.37 σ383.7
	MaTFL	121.98 σ111.8	1.12 σ1.4	43.09 σ14.6	402.82 σ327.8	630.16 σ304.6
	FedPref+FT	53.02 σ29.8	2.04 σ0.7	24.42 σ5.1	336.65 σ99.4	728.32 σ775.4
	FedPref-FT	25.82 σ11.3	2.35 σ0.5	23.85 σ4.8	337.47 σ127.0	263.87 σ179.2
Gaussian	No comm.	66.56 σ105.2	2.03 σ0.7	33.23 σ10.8	490.86 σ415.2	207.62 σ154.5
	FedAvg	1.87 σ0.7	**0.00** σ0.0	**0.00** σ0.0	2.46 σ5.4	21.98 σ9.1
	FedAvg+FT	86.31 σ87.5	0.12 σ0.3	32.64 σ53.0	225.45 σ107.7	257.47 σ164.4
	FedProx	2.70 σ2.2	0.00 σ0.0	0.00 σ0.0	**0.74** σ1.1	**20.03** σ12.0
	FedProx+FT	82.10 σ84.1	0.03 σ0.1	12.41 σ3.3	238.79 σ124.2	266.19 σ179.9
	CFL	**0.95** σ0.9	1.28 σ1.1	29.23 σ87.0	443.66 σ323.8	164.09 σ65.1
	CFL+FT	24.34 σ11.2	0.86 σ0.1	25.57 σ72.5	247.99 σ152.1	256.48 σ201.6
	MaTFL	132.91 σ126.5	1.68 σ1.4	40.85 σ17.0	327.88 σ182.6	212.49 σ91.3
	FedPref+FT	21.57 σ11.8	1.76 σ0.9	25.18 σ3.7	284.49 σ141.6	259.21 σ167.8
	FedPref-FT	23.70 σ20.4	1.88 σ0.8	27.61 σ15.6	260.01 σ108.0	123.25 σ60.7

Table 4.5: Inverted Generational Distance (IGD, ↓) metric for multi-objective solutions obtained by our proposed FedPref algorithm, compared to MaTFL, CFL, FedProx, FedAvg and individual learning without cooperation. Lower IGD means that the obtained solution set is closer to the "true" set of trade-off solutions.

		MO-LL	DMC	DST	MO-HC	MO-LLc.
Dirichlet	No comm.	45.42 σ4.4	0.41 σ0.2	1.42 σ1.0	851.67 σ307.4	55.43 σ8.1
	FedAvg	85.13 σ13.0	10.01 σ9.4	41.19 σ41.5	2737.53 σ684.5	68.28 σ6.3
	FedAvg+FT	57.25 σ9.9	0.81 σ0.2	14.52 σ25.9	733.68 σ210.4	60.43 σ14.2
	FedProx	93.19 σ15.5	9.99 σ9.3	32.71 σ38.8	2773.36 σ738.3	68.54 σ6.2
	FedProx+FT	60.71 σ11.8	0.73 σ0.2	13.47 σ26.2	**731.80** σ259.4	57.40 σ8.5
	CFL	106.68 σ15.7	0.22 σ0.1	23.57 σ34.2	804.38 σ470.8	**47.25** σ5.3
	CFL+FT	51.05 σ6.2	**0.19** σ0.1	6.30 σ0.9	733.44 σ213.2	51.04 σ3.6
	MaTFL	44.64 σ4.9	1.01 σ0.2	1.93 σ0.8	973.75 σ207.4	51.04 σ5.0
	FedPref+FT	**39.22** σ3.0	0.38 σ0.2	**0.77** σ0.3	738.36 σ183.2	47.86 σ6.7
	FedPref-FT	47.24 σ7.5	0.37 σ0.2	1.09 σ0.9	770.74 σ286.1	51.38 σ8.1
Equidistant	No comm.	57.24 σ9.4	0.40 σ0.2	0.81 σ0.4	539.53 σ141.6	76.47 σ14.9
	FedAvg	236.25 σ115.2	19.38 σ0.2	41.13 σ41.5	2463.58 σ263.7	76.65 σ6.5
	FedAvg+FT	138.72 σ29.5	2.63 σ5.5	4.83 σ0.3	**469.81** σ77.6	73.82 σ15.0
	FedProx	299.47 σ121.7	19.33 σ0.1	49.63 σ42.3	2311.35 σ574.5	83.29 σ7.9
	FedProx+FT	121.49 σ16.4	0.91 σ0.3	4.91 σ1.2	542.68 σ188.4	81.91 σ13.5
	CFL	383.86 σ99.7	0.40 σ0.2	66.43 σ39.0	504.73 σ131.9	57.72 σ10.2
	CFL+FT	87.04 σ8.7	**0.35** σ0.1	6.19 σ0.7	480.66 σ51.6	70.09 σ12.5
	MaTFL	58.39 σ4.3	1.04 σ0.2	2.16 σ1.1	931.54 σ291.1	65.97 σ11.4
	FedPref+FT	50.17 σ3.2	0.42 σ0.2	0.59 σ0.3	536.04 σ96.4	58.35 σ4.3
	FedPref-FT	**48.80** σ6.8	0.46 σ0.1	**0.48** σ0.2	540.33 σ193.3	**45.68** σ3.7
Gaussian	No comm.	42.80 σ3.4	0.39 σ0.2	0.97 σ0.5	663.02 σ294.8	52.37 σ4.2
	FedAvg	88.64 σ9.6	8.23 σ9.2	32.90 σ38.7	2853.54 σ604.7	72.67 σ4.9
	FedAvg+FT	56.38 σ11.2	0.89 σ0.3	13.71 σ26.1	817.40 σ354.0	53.55 σ5.5
	FedProx	86.44 σ17.1	10.17 σ9.3	15.74 σ25.4	2721.86 σ595.6	69.02 σ7.5
	FedProx+FT	55.36 σ5.9	0.86 σ0.3	6.16 σ1.4	816.10 σ236.0	53.48 σ7.8
	CFL	105.42 σ12.4	0.27 σ0.2	40.89 σ41.7	694.61 σ258.5	48.59 σ6.8
	CFL+FT	47.16 σ4.6	**0.17** σ0.1	6.59 σ0.6	925.00 σ816.9	50.40 σ5.3
	MaTFL	48.34 σ6.7	0.95 σ0.2	2.07 σ1.4	919.70 σ126.0	55.94 σ7.2
	FedPref+FT	**39.60** σ3.4	0.37 σ0.3	**0.71** σ0.3	**637.25** σ241.1	**46.50** σ7.4
	FedPref-FT	42.34 σ4.0	0.36 σ0.2	0.76 σ0.5	690.35 σ361.6	50.02 σ9.0

Table 4.6: Cardinality(↑) metric for multi-objective solutions obtained by our proposed FedPref algorithm, compared to MaTFL, CFL, FedProx, FedAvg and individual learning without cooperation. Higher cardinality means that a higher number of distinct trade-off solutions was found.

		MO-LL	DMC	DST	MO-HC	MO-LLc.
Dirichlet	No comm.	12.20 σ1.2	3.30 σ0.8	6.70 σ1.1	5.40 σ0.8	11.60 σ3.0
	FedAvg	8.30 σ2.8	1.00 σ0.0	1.00 σ0.0	2.90 σ1.1	9.60 σ2.2
	FedAvg+FT	10.20 σ1.3	1.10 σ0.3	3.00 σ1.6	6.50 σ1.3	9.40 σ2.2
	FedProx	9.30 σ2.4	1.00 σ0.0	1.00 σ0.0	2.90 σ0.5	10.20 σ2.6
	FedProx+FT	10.70 σ2.0	1.40 σ0.5	3.40 σ0.9	**6.80** σ1.3	9.20 σ2.1
	CFL	7.90 σ1.8	**3.60** σ0.7	2.00 σ1.3	6.00 σ0.9	12.00 σ2.9
	CFL+FT	10.10 σ1.9	3.60 σ0.7	1.90 σ0.7	6.00 σ1.1	11.20 σ2.2
	MaTFL	11.20 σ1.6	1.50 σ0.5	5.70 σ0.9	5.00 σ1.2	12.10 σ1.1
	FedPref+FT	**13.70** σ1.8	3.40 σ0.7	**7.40** σ1.0	5.80 σ1.2	12.50 σ2.1
	FedPref-FT	13.30 σ3.1	3.30 σ0.5	6.80 σ1.1	5.90 σ1.3	**12.70** σ2.7
Equidistant	No comm.	9.30 σ1.3	**3.30** σ0.5	7.40 σ1.0	6.40 σ1.1	10.80 σ1.5
	FedAvg	9.30 σ3.4	1.00 σ0.0	1.00 σ0.0	2.90 σ1.1	8.70 σ2.4
	FedAvg+FT	7.20 σ1.5	1.20 σ0.4	4.00 σ0.4	6.80 σ1.2	9.90 σ1.4
	FedProx	9.90 σ3.2	1.00 σ0.0	1.00 σ0.0	2.70 σ0.9	9.40 σ2.9
	FedProx+FT	8.40 σ1.9	1.10 σ0.3	3.80 σ0.9	**7.00** σ1.2	10.60 σ2.1
	CFL	8.20 σ1.6	3.00 σ0.6	1.20 σ0.4	6.50 σ0.8	12.00 σ2.5
	CFL+FT	9.50 σ2.2	3.30 σ0.5	2.20 σ0.4	6.80 σ0.7	10.00 σ2.1
	MaTFL	9.00 σ1.7	1.40 σ0.5	5.40 σ0.8	5.40 σ1.4	9.70 σ1.6
	FedPref+FT	**10.70** σ1.3	3.20 σ0.6	7.80 σ1.0	6.50 σ0.7	**12.30** σ1.7
	FedPref-FT	10.70 σ1.5	3.10 σ0.3	**8.00** σ0.8	6.90 σ1.0	10.00 σ2.5
Gaussian	No comm.	12.20 σ1.5	3.20 σ0.4	7.10 σ0.7	6.10 σ1.2	**12.00** σ3.3
	FedAvg	7.60 σ1.5	1.00 σ0.0	1.00 σ0.0	3.00 σ0.4	9.40 σ2.8
	FedAvg+FT	10.10 σ2.4	1.20 σ0.4	3.40 σ1.1	6.70 σ0.6	8.90 σ3.7
	FedProx	9.10 σ3.4	1.00 σ0.0	1.00 σ0.0	3.90 σ1.4	11.90 σ2.6
	FedProx+FT	10.80 σ1.8	1.10 σ0.3	3.00 σ1.0	6.80 σ1.3	10.00 σ3.0
	CFL	6.90 σ3.0	3.40 σ0.8	1.20 σ0.4	5.50 σ1.6	11.80 σ2.6
	CFL+FT	12.00 σ2.5	**3.90** σ0.3	1.80 σ0.6	6.70 σ1.3	11.20 σ2.3
	MaTFL	11.70 σ1.7	1.70 σ0.6	5.80 σ1.2	5.10 σ1.1	11.90 σ2.1
	FedPref+FT	**12.60** σ2.0	3.30 σ0.8	7.10 σ0.5	6.30 σ1.1	11.80 σ2.4
	FedPref-FT	12.50 σ2.2	3.30 σ0.5	**7.80** σ1.3	**7.00** σ1.1	10.70 σ1.5

Table 4.7: Experimental results comparing multi-objective metrics obtained by the individual components of our algorithm.

		Hypervolume ↑	Cardinality ↑	Sparsity ↓	IGD ↓
Clustering+FT	MO-LL	187.16 σ11.1($\cdot 10^7$)	**13.70** σ2.1	17.12 σ6.5($\cdot 10^2$)	42.12 σ4.3
	DMC	174.08 σ16.7	2.60 σ0.5	1.17 σ0.2	0.37 σ0.1
	DST	211.92 σ24.9($\cdot 10^1$)	7.20 σ1.2	28.99 σ11.5	1.47 σ1.2
	MO-HC	417.69 σ51.5($\cdot 10^4$)	5.90 σ1.1	434.17 σ175.2($\cdot 10^4$)	819.40 σ270.3
	MO-LLc.	97.95 σ6.4($\cdot 10^7$)	12.20 σ3.1	242.55 σ153.5($\cdot 10^1$)	50.27 σ5.8
Clustering-FT	MO-LL	194.61 σ7.6($\cdot 10^7$)	11.70 σ2.9	19.35 σ14.5($\cdot 10^2$)	49.28 σ11.5
	DMC	166.12 σ30.5	2.70 σ0.8	1.46 σ0.8	0.43 σ0.2
	DST	**222.85** σ4.9($\cdot 10^1$)	6.80 σ1.3	43.30 σ33.4	0.99 σ0.8
	MO-HC	411.80 σ41.3($\cdot 10^4$)	6.00 σ1.4	500.84 σ373.0($\cdot 10^4$)	810.57 σ276.1
	MO-LLc.	97.22 σ10.6($\cdot 10^7$)	11.30 σ2.8	158.75 σ185.1($\cdot 10^1$)	54.90 σ6.7
Weighted agg.+FT	MO-LL	175.09 σ14.6($\cdot 10^7$)	12.40 σ1.9	77.43 σ95.1($\cdot 10^2$)	50.04 σ6.2
	DMC	145.41 σ28.5	2.30 σ0.5	8.63 σ18.2	0.74 σ0.2
	DST	180.06 σ27.7($\cdot 10^1$)	4.10 σ1.3	21.83 σ23.5	4.02 σ1.5
	MO-HC	426.41 σ20.7($\cdot 10^4$)	6.60 σ1.2	269.32 σ107.9($\cdot 10^4$)	**679.38** σ205.9
	MO-LLc.	98.90 σ4.7($\cdot 10^7$)	10.40 σ2.0	285.12 σ145.8($\cdot 10^1$)	50.11 σ4.2
Weighted agg.-FT	MO-LL	186.64 σ13.9($\cdot 10^7$)	8.80 σ2.7	**5.86** σ7.2($\cdot 10^2$)	75.88 σ16.7
	DMC	59.68 σ29.9	1.10 σ0.3	**0.18** σ0.5	1.11 σ0.2
	DST	157.11 σ22.6($\cdot 10^1$)	2.20 σ1.2	**12.85** σ14.2	5.98 σ1.8
	MO-HC	300.40 σ110.3($\cdot 10^4$)	**7.20** σ1.2	**202.30** σ311.4($\cdot 10^4$)	1461.76 σ522.8
	MO-LLc.	93.47 σ7.7($\cdot 10^7$)	9.40 σ3.5	**51.86** σ45.9($\cdot 10^1$)	64.97 σ4.9
FedPref+FT	MO-LL	193.94 σ7.5($\cdot 10^7$)	13.70 σ1.8	19.35 σ8.6($\cdot 10^2$)	**39.22** σ3.0
	DMC	202.79 σ29.7	**3.40** σ0.7	1.83 σ1.0	0.38 σ0.2
	DST	220.36 σ10.3($\cdot 10^1$)	**7.40** σ1.0	28.00 σ7.6	**0.77** σ0.3
	MO-HC	421.43 σ39.5($\cdot 10^4$)	5.80 σ1.2	408.45 σ167.6($\cdot 10^4$)	738.36 σ183.2
	MO-LLc.	100.04 σ5.5($\cdot 10^7$)	12.50 σ2.1	194.61 σ88.7($\cdot 10^1$)	**47.86** σ6.7
FedPref-FT	MO-LL	**201.97** σ7.0($\cdot 10^7$)	13.30 σ3.1	19.38 σ17.0($\cdot 10^2$)	47.24 σ7.5
	DMC	**203.74** σ13.2	3.30 σ0.5	1.88 σ0.8	**0.37** σ0.2
	DST	218.04 σ18.3($\cdot 10^1$)	6.80 σ1.1	33.07 σ7.9	1.09 σ0.9
	MO-HC	**435.87** σ36.5($\cdot 10^4$)	5.90 σ1.3	517.10 σ377.2($\cdot 10^4$)	770.74 σ286.1
	MO-LLc.	**101.57** σ8.8($\cdot 10^7$)	**12.70** σ2.7	92.45 σ68.8($\cdot 10^1$)	51.38 σ8.1

4.5 | Summary and outlook

In this chapter, we have discussed a first preference-based multi-solution algorithm for federated multi-objective learning. In Section 4.1, we have presented the concept of preference heterogeneity and the motivation for choosing a personalised design for the federated algorithm. This algorithm, based on a combination of similarity-based recursive clustering and weighted aggregation, was discussed in Section 4.2. This algorithm preserves the privacy of clients: it is capable of functioning using only the respective client model updates; no further information about client objectives is required.

We have validated the performance of our algorithm on multiple and varied problems and preference distributions, comparing it to classical benchmarks as well as other heterogeneity-mitigating algorithms. We have analysed the results from two different points of view: First, we have considered the traditional client-centric view in Section 4.3, demonstrating that our algorithm outperforms the alternatives in many cases in terms of mean client performance, and represents a reliable choice in all others. Further experiments were carried out to study the characteristics of the algorithm.

In addition, we have discussed a multi-objective view of the federated system in Section 4.4, analysing the performance of the FedPref algorithm under multi-objective aspects. Our results show that, while the algorithm does not currently explicitly enforce multi-objective characteristics, it nevertheless performs well on several common multi-objective metrics. These results, too, persist across different types of problems and heterogeneity distributions, in contrast to the compared algorithms.

In the following chapter, we discuss the construction of additional benchmarking problems for the federated multi-objective setting to validate FMOL algorithms on more commonly used types of problems.

5 | A new class of benchmarks for federated multi-objective learning

Contents

Progress in FMOL critically depends on benchmarks that represent genuine conflicts between objectives. Existing benchmarks are primarily based on multi-task classification problems, where tasks can often be optimised jointly without inherent conflict, as we demonstrate in this chapter. As a result, they fail to represent the full spectrum of difficulty in multi-objective problems, limiting their utility for evaluating FMOL algorithms. Federated Multi-objective Learning has only recently emerged as a dedicated direction of research, with relatively few works exploring general solution algorithms [Ask24; Har25a; Yan23b]. To the best of our knowledge, the benchmarking of FMOL algorithms has not yet been explicitly addressed in the literature. In the absence of dedicated benchmarks, a common practice in the Federated Learning domain is to re-purpose existing ones from the centralised setting. In this vein, several works on FMOL [Ask24; Yan23b] employ Multi-Task datasets, originally designed for benchmarking centralised multi-task learning (MTL) and multi-objective learning algorithms, to evaluate their methods.

One of the most commonly used datasets, both in federated and non-federated settings, is Multi-MNIST [Sen18]. Multi-MNIST is constructed by combining two overlapping MNIST digits at an offset into a single image and concatenating the associated labels into an ordered sequence. The two tasks consist of classifying the left-most and right-most digit, respectively. Similar datasets constructed analogously include Multi-Fashion

[Lin19], combining two Fashion-MNIST images; Fashion-MNIST [Lin19], combining samples from MNIST and Fashion-MNIST, respectively; and CIFAR-MNIST [Cho20], combining CIFAR10 and MNIST images. In addition to such newly constructed datasets, pre-existing multi-label classification benchmarks such as the CelebA dataset [Liu15] have also been used. In such works, straightforward learning approaches demonstrated an apparent trade-off between tasks.

A handful of works use other types of multi-objective problems for validation. Kinoshita et al. [Kin24] focus on solving unsupervised multi-objective optimisation problems such as clustering, which does not readily extend to general FL scenarios. Hartmann et al. [Har25a] use existing multi-objective reinforcement learning (MORL) benchmarks. While these represent problems that have definite, intuitively verifiable inherent trade-offs, sequential learning problems remain understudied in FL and thus should not be considered generally representative of the problem space.

Finally, some works also confront domain-specific or otherwise more narrowly defined problems that are multi-objective, without considering the general applicability. Fair federated learning is a line of research focused on ensuring fairness both between clients [Hu22a; Ju24], and in the resulting model [Meh22]. Though this abstraction is not remarked upon, the latter problem is multi-objective, as fairness and accuracy are known to conflict on biased datasets. None of these domain-specific problems present an immediately compelling alternative to the established MTL problems for general benchmarking purposes. However, we argue in the next section that the sole reliance on MTL problems is nevertheless suboptimal, as this class of benchmarks is likely not representative of the full difficulty of FMOL.

This chapter makes three key contributions:

- **Benchmark analysis:** We show that widely used multi-task benchmarks fail to exhibit the expected trade-offs under federated settings, even when applying simple methods such as FedAvg.
- **Fairness-based benchmarks:** To address this limitation, we introduce an alternative class of FMOL benchmarks built upon well-established fairness metrics from the fair ML literature. These benchmarks naturally encode conflicting objectives while remaining simple and flexible to construct and apply.
- **Empirical validation:** Through experiments with both baseline and state-of-the-art FMOL algorithms, we demonstrate that fairness-based benchmarks reveal genuine multi-objective behaviour, thereby providing a more meaningful test-bed for algorithmic evaluation.

5.1 | Multi-Task Benchmarks in Federation

In this section, we demonstrate the observation that served as the motivation for this work: that in certain natural settings, solving multi-task problems in federation appears to reduce or remove the conflict between individual tasks, simplifying the problem considerably from a multi-objective perspective. Conceptually, the difficulty of solving multi-objective problems arises from conflict between the individual objectives, where optimising one objective

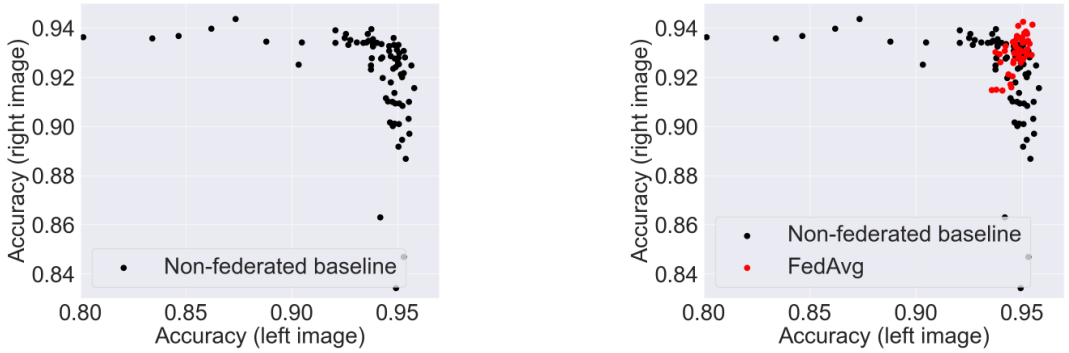


Figure 5.1: Results for non-federated (left) and with federated experiments (right) on Multi-MNIST with heterogeneous fixed preferences. Non-federated results show an apparent trade-off between the two objectives, but federated results do not. Federated results outperform non-federated ones, despite the forced collaboration between clients with different objective preferences.

reduces the utility of another. To accurately assess the performance of multi-objective algorithms, benchmarking problems should reflect this challenge. In some domains, the inherent conflict between objectives is immediately obvious. For example, a route planning algorithm for an autonomous vehicle may be expected to both minimise fuel usage and minimise the travel time to a given destination. Both objectives cannot generally be satisfied at the same time, as travelling faster consumes more fuel. In other domains, however, determining whether conflict between objectives is inherent or not is much more difficult. This is the case for the classification problems that are most often considered in FL. In multi-task benchmarks, non-federated experiments have shown a trade-off between the two objectives when assigning different fixed preferences to each, indicating that improvement in one objective harms the other. Yet it is not immediately clear that this is caused by an inherent conflict between the objectives. The model architecture used to solve such problems typically consists of a shared block, followed by individual model segments for each task. Given this architecture, and the independent nature of the two parallel classification tasks, there is no apparent reason why a sufficiently expressive network should not be able to separate the tasks and so satisfy both. Indeed, we speculate that the observed trade-off behaviour may be a limitation of the learning algorithm caused by a lack of exploration of the parameter space, not a characteristic of the underlying learning problem. While the general absence of a conflict in MTL benchmarking datasets is difficult to prove, we give a motivating example that shows the apparent collapse of the Multi-MNIST benchmark in a particular federated use case: the preference-heterogeneous setting.

Preference heterogeneity has not yet received much attention in the literature, but is nonetheless a natural setting. It may occur in any use case where clients are self-interested: when training personalised user recommender systems, foundation models on proprietary data across multiple enterprises, or on-line route planners on autonomous vehicles. In this setting, each participating client has its own preferences regarding the importance of

individual objectives. On problems with conflicting objectives, we would expect this to cause complications in the federated aggregation step: diverging local training trajectories may be difficult to reconcile. However, we observe a different result: In our experiments, the non-federated baseline does reproduce a set of trade-off solutions, or *Pareto front*, but the FedAvg algorithm yields better results with far less apparent trade-off. FedAvg is not known for handling heterogeneity well; yet this type of heterogeneity on this problem appears to improve the output significantly, removing conflict between objectives. We speculate that federated preference heterogeneity has the same effect as intentionally varying preference weights during the learning process, an approach that is employed intentionally in the design of more sophisticated algorithms solving (non-federated) MTL. A notable example is [Sen18], where such an algorithm generates a (single, arbitrary) solution that dominates a Pareto front generated with fixed weights – similar to what we observe here. From these considerations, we conclude that standard MTL benchmarks may not be a challenging benchmark for FMOL algorithms. While the general absence of a conflict in this and other MTL benchmarking datasets is difficult to prove, we consider this argument compelling enough to propose the use of additional classes of benchmarks in combination with MTL datasets. In the remainder of this chapter, we propose a different class of multi-objective (multi-criteria) problems as benchmarks and, using these problems, demonstrate that MOO remains a challenge in federation.

5.2 | Designing Alternative Benchmarks – Group Fairness

Based on the observations outlined in the previous section, we argue that more challenging benchmarking problems are needed to comprehensively evaluate the performance of FMOL algorithms. To address this need, we introduce a new class of benchmarks constructed by adapting established problems from the field of fair machine learning into generally applicable problem formulations. This section first outlines key concepts in fair machine learning before detailing our proposed benchmarks.

5.2.1 | Background: Fairness in Machine Learning

Many real-world datasets, particularly those involving demographic data, are known to contain imbalances that reflect underlying cultural or societal biases. Examples include racial disparities in criminal sentencing decisions, gender-based differences in income determination, and age-related biases in health records. Training prediction models on such datasets risks propagating these biases, leading to discriminatory behaviour in automated decision-making systems. A well-known example is the COMPAS dataset used for recidivism prediction, which has been shown to systematically discriminate against Black defendants [Ang22].

Bias mitigation has been extensively studied, with existing approaches typically categorised into three families: debiasing the underlying dataset (pre-processing), preventing the learning of biases during training (in-processing), and modifying the output of the trained model to enhance fairness (post-processing) [Meh21]. Of particular interest here are in-processing methods that can formulate the learning process as a multi-objective problem, introducing fairness as additional objectives. This formulation serves as the foundation for our proposed benchmarks.

Fairness in machine learning has been formalised through various metrics, often formulated with respect to a binary *sensitive attribute*, quantifying disparities in predicted outcomes between subpopulations [Meh21]. A classifier is considered perfectly fair if outcomes are statistically indistinguishable across these groups. Such group fairness metrics include demographic parity [Dwo12; Kus17], equality of opportunity [Har16], and equalised odds [Har16].

Demographic parity (DP) requires the overall probability of a positive classification outcome, such as loan approval, to be equal between the in-group and the out-group. Let X be the set of input data, Y the set of labels and S the labels of sensitive attributes. In formal terms, a classifier satisfies demographic parity if, across all predictions,

$$P(\hat{y} = 1|s = 0) = P(\hat{y} = 1|s = 1), \tag{5.1}$$

where $\hat{y}$ is the binary predicted outcome, and $s \in S$ is the sensitive attribute.

Equality of opportunity (EO) demands equal probabilities of *true* positive outcomes between groups, i.e.

$$P(\hat{y} = 1|s = 0, y = 1) = P(\hat{y} = 1|s = 1, y = 1), \tag{5.2}$$

where $y \in Y$ is the ground-truth label of a given sample.

Equalised odds (EOD) requires equal probabilities of true positive as well as false positive outcomes across groups:

$$\begin{aligned} P(\hat{y} = 1|s = 0, y = 1) = P(\hat{y} = 1|s = 1, y = 1) \\ \wedge P(\hat{y} = 1|s = 0, y = 0) = P(\hat{y} = 1|s = 1, y = 0). \end{aligned} \tag{5.3}$$

For practical use as a fairness score on classification data, these definitions can be reformulated as the stochastic difference between the left- and right-hand side of the equation, e.g. for DP, we formalize the Difference of Demographic Parity (DDP) as follows:

$$DDP(X, Y, S, f) = \frac{1}{n_{s=0}} \sum_{\substack{x \in X \\ s=0}} [f(x) > 0.5] - \frac{1}{n_{s=1}} \sum_{\substack{x \in X \\ s=1}} [f(x) > 0.5], \tag{5.4}$$

where $f : X \to [0,1]$ is the predictor, $[f(x) \to 0,1]$ is the binary classification decision and $n_{s=s'}$ is the number of samples with sensitive attribute value $s' \in \{0,1\}$. The existence of *fairness impossibility* is a well-known result in fair machine learning, stating that certain fairness concepts, including demographic parity, cannot be jointly optimised with error-based metrics on biased datasets. A full overview of existing fairness metrics, together with a comprehensive discussion of theoretical and empirical incompatibility results, can be found in [Pes22].

5.2.2 | Formulation of benchmarking problem

Using these well-established metrics, we construct straightforward additional benchmarks for FL. We focus on ease of implementation and evaluation, discarding more subtle design choices in favour of simplicity.

5.2.2.1 | Objectives

The stochastic formulation of the fairness metrics described above is not differentiable, and as such not well-suited for direct use as a loss function in stochastic gradient descent. Various relaxation approaches have been proposed, e.g. [Cel19; Loh20], often in combination with specific solution algorithms such as constraint-based optimization. Two other works, focused specifically on fair federated learning, side-step the problem by using fairness metrics only as a secondary scoring mechanism[Meh22] or optimization constraint [Cui21]. None of these formulations generalises readily into an abstract multi-objective problem that admits different solution approaches, such as stochastic multi-gradient descent. Instead, we propose to use a recently introduced relaxation method that yields a differentiable approximation of the metric [Pad21], directly usable as a loss function. The hyperbolic tangent relaxation is straightforwardly applicable to most standard group fairness metrics, including those presented here, is model-agnostic, and can be be used for multiple metrics simultaneously. This gives us scalability, allowing the design of many-objective problems, and flexibility in choosing the local learning approach. Under the tanh relaxation, the prediction $f(x) : X \to [0,1]$ of the classifier is relaxed to

$$\hat{f}(x) = tanh(c \cdot max(0, 2f(x) - 1))/2 + 0.5, \tag{5.5}$$

where $c \in \mathbb{R}$ regulates the trade-off between the precision of the approximation and the behaviour of the gradient[1]. The relaxed prediction is then used in place of the binary result in computing the chosen fairness metric, e.g. for the DDP metric:

$$\widehat{DDP}(X, Y, S, \hat{f}) = \frac{1}{n_{s=0}} \sum_{\substack{x \in X \\ s=0}} \hat{f}(x) - \frac{1}{n_{s=1}} \sum_{\substack{x \in X \\ s=1}} \hat{f}(x) \tag{5.6}$$

We use the relaxed gap metric for one or more fairness metrics as individual objectives in a multi-objective problem, combined with the accuracy objective. (For the accuracy objective, an appropriate loss function for the learning problem is used.) Note that many fairness benchmarking datasets have more than one potentially sensitive attribute, e.g. gender and race, with attributes mutually independent. This provides a straightforward avenue for the construction of problems with more than two objectives. Similarly, multiple different fairness metrics can be applied simultaneously as separate objectives, with various group fairness metrics known to mutually conflict [Cas22].

Another benefit of this fairness formulation is flexibility w.r.t. the local learning strategy. Some FMOL algorithms specify a local learning algorithm, e.g. multi-gradient descent [Ask24; Yan23b]; others that are based on model similarity, e.g. [Har25a], do not. This formulation of the fairness problem is equally accessible to all these algorithms, and even allows the testing of algorithms not specific to the setting, e.g. the FedAvg baseline algorithm [McM17d].

5.2.2.2 | Datasets

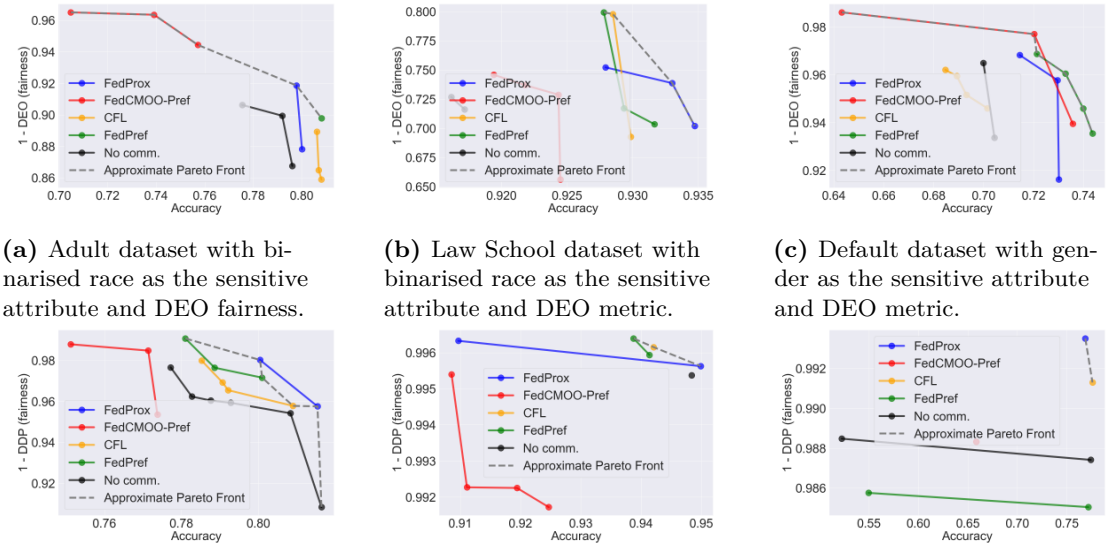
A great number of different benchmarking datasets in varying size and shape exist in the domain of fair machine learning. All such datasets that contain sensitive attributes can

1 This definition differs slightly from that given in the reference paper, which was based on predictions in the range of [-1, 1].

Table 5.1: Selection of common benchmarking datasets in the fair machine learning domain, all usable with our proposed formulation as drop-in benchmarks for FMOL algorithms.

Dataset	Description	Sensitive attrs.
Adult Income [Bec96]	Demographic data, predicting binary income class	Gender, Race
Law School [Wig98]	Demographic and academic data, predicting bar passage	Gender, Race
Credit Default [Yeh09]	Demographic and financial data, predicting credit card default	Gender, Age
Compas [Ang22]	Demographic and criminal history data, predicting recidivism	Race
CelebA [Den19; Liu15]	Multi-label image classification of faces	Gender
Heritage Health [Gol11] [Gol11]	Demographic and health data, predicting hospitalization	Age

be used with the loss functions defined here. The vast majority is based on real-world data collected for other purposes, with underlying biases identified by later research. This is an advantage for benchmarking, as datasets represent tangible and realistic use cases, many with obvious relevance to the federated setting. Though this real-world origin can also present challenges, such as flawed or incomplete data, many frequently used datasets are available in cleaned form. We list a selection of commonly used datasets in Table 5.1. For a more comprehensive overview of datasets we refer to the appendix of the survey by Pessach et al. [Pes22]; a second survey [Le 22] contains a particularly detailed description of several datasets.



(a) Adult dataset with binarised race as the sensitive attribute and DEO fairness.

(b) Law School dataset with binarised race as the sensitive attribute and DEO metric.

(c) Default dataset with gender as the sensitive attribute and DEO metric.

(d) Adult dataset with binarised race as the sensitive attribute and DDP metric.

(e) Law School dataset with binarised gender as the sensitive attribute and DDP metric.

(f) Default dataset with gender as the sensitive attribute and DDP metric.

Figure 5.2: Results of different algorithms on a selection of benchmark problems for accuracy and equality of opportunity (top row) and accuracy and demographic parity (bottom row). All clients were assigned the same preferences during a run, with 10 runs performed on preferences from $(0., 1.0)$ to $(0.9, 0.1)$, modified by steps of $(+0.1, -0.1)$. Each point represents the mean client output for a single run, with the Pareto fronts across all runs reported for each algorithm. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

5.3 | Experiments

We demonstrate the validity and usability of the proposed class of fairness benchmarks by constructing ten bi-objective example problems, combining three different fairness datasets with a total of five sensitive attributes with two different fairness metrics. We select three common and readily available fairness benchmarking datasets:

- **UCI Adult**: Data extracted from the 1994 US Census database. Using demographic information to predict whether a person's income exceeds \$50,000 per year. The sensitive attributes used in our experiments are gender and race (binarised into *white* and *not white*).
- **Law School**: Data on US law students between 1991-1997. Using demographic information and earlier test scores to predict whether a candidate passes the bar. Sensitive attributes used are binarised race and gender.
- **Credit Card Default**: predicting from personal information and credit card history whether a bank customer will default on their next credit card payment. The sensitive attribute is the gender of the customer.

We use accuracy as one objective metric, and either the difference of demographic parity

(DDP) or difference of equality of opportunity (DEO) as fairness metric to construct two distinct bi-objective problems on each dataset-attribute combination, defining the fairness loss as described in the previous section and the accuracy loss as binary cross-entropy with logits.
On these benchmarks, we run four representative FL algorithms: FedProx, CFL, FedCMOO, and FedPref. FedProx is a standard baseline algorithm performing centralised aggregation, with some tolerance for client heterogeneity [Li20a]. CFL is a clustering-based algorithm that generates personalised client models [Sat19]. Though originally designed for settings with incompatible client data, the adaptive clustering strategy may be applicable for multi-objective heterogeneity as well. FedCMOO is a recent algorithm designed specifically for federated multi-objective learning [Ask24]. We implement the FedCMOO-Pref variant, equipped to handle homogeneous objective preferences. In contrast, FedPref, another recent FMOL algorithm, is intended specifically for federating preference-heterogeneous clients [Har25a]. Finally, we also run the standard non-federated baseline. All algorithms are tuned via grid search, with details reported in the appendix.
We test two natural and distinct scenarios for federating multi-objective problems. First, we solve a multi-objective problem in collaboration between clients that all share the same objective preferences. Then, we run the setting where all clients have individual, heterogeneous objective preferences. The experiments reported here are run on systems of 10 clients; the appendix includes further experiments on up to 50 clients.

5.3.1 | Homogeneous preferences

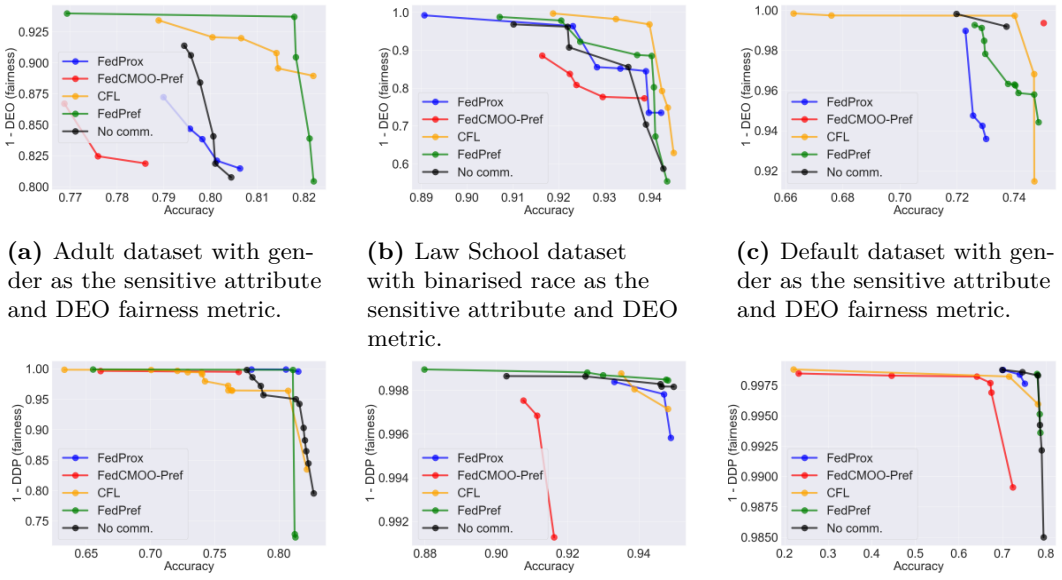
This setting corresponds to the multi-objective equivalent of the most common focus in FL, where clients collaborate to train a single global model that generalises over all data available in distribution. To explore the multi-objective performance of algorithms on this benchmark, we generate a set of 10 equally-spaced preference weights. We run each benchmarking problem 10 times, once for each preference weight, and report the mean client results for each run. Following standard practice from multi-objective optimization, we compute the Pareto front of solutions per algorithm, and report the hypervolume metric for each. We also show the minimum and maximum values found for each objective and algorithm in the appendix, illustrating the spread of results.
Fig. 5.2 shows the Pareto fronts obtained for six of the benchmarking problems – three optimising the DEO fairness metric, and three the DDP metric, with the corresponding hypervolume values reported in Tables 5.2 and 5.3. A trade-off between the accuracy and fairness objective is readily apparent in all six plots, with different performances by the tested algorithms on different datasets. Nevertheless, some general observations can be noted: in almost all cases, all federated algorithms outperform the non-federated baseline. (An exception is the DDP metric on the Law School dataset, where the single solution reported for the baseline is most likely an outlier that did not converge. Statistical noise is a challenge for these datasets that is discussed in more detail at the end of this section.) The FedProx algorithm performs relatively well in this setting, indicating that, at least in the absence of objective heterogeneity, the basic algorithm is capable of finding some appropriate trade-off solutions. The other federated algorithms all show mixed performances: in Fig. 5.2(a) and Fig. 5.2(c), FedCMOO notably explores sections of the

Pareto front discovered by no other algorithm. A similar tendency, though less successful, appears in the Adult dataset with the DDP metric (Fig. 5.2(d)). The increased exploration range may be explained by the design of FedCMOO, which adaptively adjusts the initial objective preferences during the training process. The FedPref algorithm discovers at least one Pareto-dominant solution in five cases, but appears occasionally very limited in its exploration of the Pareto front (see e.g. Fig. 5.2(a) and 5.2(e)). It is possible that the clustering mechanism of the algorithm is counterproductive in this homogeneous setting, where no obvious groupings of clients exist. A similar issue may be limiting the performance of the CFL algorithm by separating clients where no inherent incompatibility exists. Finally, it should be noted that these results may not represent the true potential of each algorithm. In a thorough multi-objective evaluation of the algorithms, a heuristic would normally be employed to search the space of preference weights to generate a balanced Pareto front. In this work, intended mainly to demonstrate the usability of the proposed class of benchmarks, the exploration was instead restricted to a predefined set of preference weights.

5.3.2 | Heterogeneous preferences

This second setting represents a use case with high heterogeneity, where each client in the federation has individual preferences. Such a setting is commonly connected to Personalised Federated Learning (PFL) approaches, where the focus of the algorithm is shifted from generalised global performance to individual client performance. Instead of generating a single global model, the traditional aggregation approach is modified to yield a separate personalised model for each client, fitted to that client's unique characteristics.
We run the same selection of algorithms as in the previous experiments. As some of these algorithms are not designed to generate personalised client models, we have included the option of a single fine-tuning step at the end of the training phase in the hyperparameter tuning. 10 preference sets were generated uniformly at random and submitted to each algorithm. We visualise the results in Fig.5.3 and report the corresponding hypervolumes in Tables 5.4 and 5.5, with additional results, including for a greater number of clients, reported in the appendix. Unlike in the homogeneous setting, we do not average the results for each run, but instead consider the individual client solutions. Due to space limitations, the min-max analysis is once again included in the appendix.
We observe that the FedProx algorithm performs notably worse in the heterogeneous setting, particularly in experiments with the DEO metric – see e.g. its performance on the Adult dataset (Fig. 5.3(a)), where it is outperformed even by the non-federated baseline. This is consistent with our expectation that algorithms with centralized aggregation would struggle in preference-heterogeneous settings with conflicting objectives. The FedCMOO algorithm, too, is not designed for this setting, and perhaps has difficulties in reconciling incompatible clients. FedPref and CFL, the personalized algorithms, both do better in this setting. For both the homogeneous and the heterogeneous setting, we also observe that the behaviour observed on the different datasets is quite consistent as the number of clients increases, as seen in the additional results reported in the appendix. Finally, we note that the DDP metric generally appears more difficult to solve than the DEO metric, with fewer points discovered on the Pareto front, and a tendency for those points to be

extreme. The trade-off between DDP and accuracy may be more difficult to regulate, or learning trajectories diverge earlier, allowing less time for collaboration between clients. This hypothesis may explain why even the more successful FL algorithms struggle to outperform the non-federated baseline in Fig. 5.3, and why there is relatively little diversity in the Pareto fronts discovered in the homogeneous setting in Fig.5.2.



(a) Adult dataset with gender as the sensitive attribute and DEO fairness metric.

(b) Law School dataset with binarised race as the sensitive attribute and DEO metric.

(c) Default dataset with gender as the sensitive attribute and DEO fairness metric.

(d) Adult dataset with gender as the sensitive attribute and DDP fairness metric.

(e) Law School dataset with gender as the sensitive attribute and DDP metric.

(f) Default dataset with gender as the sensitive attribute and DDP metric.

Figure 5.3: Results of different algorithms on a selection of benchmark problems for accuracy and equality of opportunity (top row) and accuracy and demographic parity (bottom row). Clients were assigned heterogeneous preferences during each run, generated uniformly at random but the same across algorithms. Each point represents the output of a single client, with the Pareto fronts across all runs reported for each algorithm. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

Table 5.2: Hypervolumes of global performance results for accuracy and DEO on homogeneous preferences. Higher is better (Fairness metrics are inverted, as in the results figures). Only results from the algorithm-specific Pareto front are reported (see also Fig. 5.2)

	Accuracy - DEO				
Data - Sensitive Attr.	FedProx	CFL	FedCMOO	FedPref	no comm.
Adult - Gender	**0.701**	0.687	0.676	0.696	0.678
Adult - Race	**0.735**	0.719	0.730	0.726	0.721
Law School - Gender	0.882	0.821	**0.919**	0.836	0.796
Law School - Race	0.703	0.742	0.689	**0.744**	0.667
Default - Gender	0.707	0.675	**0.724**	0.720	0.680

Finally, a comparison of the visualised Pareto fronts with the corresponding hypervolume and min-max values reveals an interesting insight: while the potential values of the objective

Table 5.3: Hypervolumes of global performance results for accuracy and DDP on homogeneous preferences. Higher is better (Fairness metrics are inverted, as in the results figures). Only results from the algorithm-specific Pareto front are reported (see also Fig. 5.2)

	Accuracy - DDP				
Data - Sensitive Attr.	FedProx	CFL	FedCMOO	FedPref	no comm.
Adult - Gender	0.763	0.788	0.754	0.761	**0.805**
Adult - Race	**0.799**	0.792	0.764	0.793	0.796
Law School - Gender	**0.946**	0.938	0.920	0.938	0.944
Law School - Race	0.911	0.944	**0.945**	0.928	0.940
Default - Gender	0.764	**0.769**	0.651	0.760	0.765

metrics cover the same interval, in practice the trade-off between the objectives plays out in different magnitudes on the two axes. Hence e.g. the highest overall hypervolume on the Adult dataset with gender as the sensitive attribute in Table 5.5 is achieved by the no-communication baseline, even though it does not appear obviously superior in the illustration in Fig. 5.3(d). This imbalanced magnitude of metrics presents a challenge that is often encountered in the real world. As such, there is use in evaluating the ability of algorithms to cope with such problems, where the magnitude of objective gradients may differ.

Table 5.4: Hypervolumes of global performance results for accuracy and DEO on heterogeneous preferences. Higher is better (Fairness metrics are inverted, as in the results figures). Only results from the algorithm-specific Pareto front are reported (see also Fig. 5.3).

	Accuracy - DEO				
Data - Sensitive Attr.	FedProx	CFL	FedCMOO	FedPref	no comm.
Adult - Gender	0.703	0.767	0.681	**0.772**	0.734
Adult - Race	0.715	0.794	0.732	0.796	**0.802**
Law School - Gender	0.942	**0.946**	0.896	0.939	0.940
Law School - Race	0.931	**0.941**	0.829	0.929	0.909
Default - Gender	0.722	**0.745**	0.745	0.742	0.736

Table 5.5: Hypervolumes of global performance results for accuracy and DDP on heterogeneous preferences. Higher is better (Fairness metrics are inverted, as in the results figures). Only results from the algorithm-specific Pareto front are reported (see also Fig. 5.3).

	Accuracy - DDP				
Data - Sensitive Attr.	FedProx	CFL	FedCMOO	FedPref	no comm.
Adult - Gender	0.814	0.816	0.766	0.811	**0.822**
Adult - Race	0.822	0.818	0.773	**0.830**	0.823
Law School - Gender	0.947	0.947	0.914	0.947	**0.948**
Law School - Race	0.949	0.949	0.949	0.949	**0.949**
Default - Gender	0.750	0.780	0.722	0.786	**0.793**

5.3.3 | Practical considerations

Our implementation of these experiments will be available on git[1]. Deploying this class of benchmarks in other existing implementations of federated algorithms requires minimal modifications in principle. Many fairness datasets are widely available and, as with those used here, can often be accessed directly through common machine learning libraries such as PyTorch. The local learning process does not generally need adjustment beyond the addition of the fairness loss functions and evaluation metrics, which are lightweight and easily portable. Tuning local learning parameters to explore diverse trade-offs can be challenging; we provide notes on parameter selection and implementation details in the appendix.

5.4 | Summary

In this chapter, we have introduced a new class of benchmarks for evaluating Federated Multi-objective Learning algorithms, addressing the limitations of existing multi-task benchmarks that often lack genuine objective conflicts in federated settings. Conflicts between utility and fairness, as well as between different fairness metrics, are well-established in the domain of Fair Machine Learning. Our experiments confirm that our proposed fairness-based benchmarks are versatile, simple to implement, and capable of exposing meaningful trade-offs between objectives across various FL scenarios.

Rather than advocating for a complete replacement of current benchmarks, we argue for their diversification to better reflect the challenges of FMOL. Future work should investigate improved data partitioning strategies to reduce noise in fairness datasets and better control client heterogeneity.Finally, our findings suggest that heterogeneous client preferences may, paradoxically, facilitate optimization by improving parameter space exploration in multi-task FL. Understanding and exploiting this effect could open new directions for distributed multi-task and multi-objective learning.

1 The code is not yet publicly available, as this work remains under double-blind review at a conference. It will be published alongside the paper.

6 | Towards real-world applications in the aerospace domain

Contents

With advances in hardware and software capabilities, distributed satellite mission configurations are progressively replacing the classical paradigm of individual monolithic spacecraft. As even small satellites become capable of generating, storing and processing increasingly large amounts of data through various on-board sensors, downlink capacity is emerging as a major bottleneck in processing the gathered information. To manage this problem, there is an ongoing drive towards shifting data processing onto satellites[Izz22] – this strategy is referred to as Orbital Edge Computing (OEC)[Den20]. The overarching idea of OEC is to leverage on-board computing capabilities of each satellite to process locally gathered data, reducing the size and amount of required transmissions and speeding up evaluation and decision-making. This could, for example, enable networks of Earth observation satellites to rapidly identify and flag potential wildfires, or allow communication satellites to autonomously adjust capacity based on changing demand patterns.

In addition to isolated on-board learning approaches on single satellites, another promising line of research proposes deploying Federated Learning [McM17a] (FL) across multiple satellites. This could permit the joint training of on-board machine learning models across the data gathered by multiple satellites while supporting a limited communication budget [Jab23]. Under a FL scheme, each satellite would perform on-board machine learning on the data it collects, training a local model – step (1) in Fig. 6.1. These models would be shared periodically among participants (5) or with a ground-based server (2), allowing them to be aggregated into a more accurate global model (3) on which to continue training. Aggregation could take place with the aid of a parameter server on the ground or in orbit, or in a fully distributed manner between satellites. Fundamental advantages of this approach include a vastly reduced communication cost compared to the transmission of raw data, and the inherent privacy advantages of compartmentalising data on satellites.
Current literature on the use of Federated Learning in Orbital Edge Computing is focused primarily on a single use case: using Federated Learning in a single, dedicated constellation of satellites. However, another natural scenario appears largely unstudied: the potential for satellites from different missions and providers to form (ad-hoc) cross-provider collaborations. This scenario is interesting in itself as a near-term use case, while also being representative of the challenges that may be encountered in the design of more complex future missions. Satellite swarm-based deep space exploration, federated spacecraft designs, or distributed autonomous space debris mitigation systems could all benefit from effective collaborative learning techniques. The remainder of this chapter is divided into two parts:

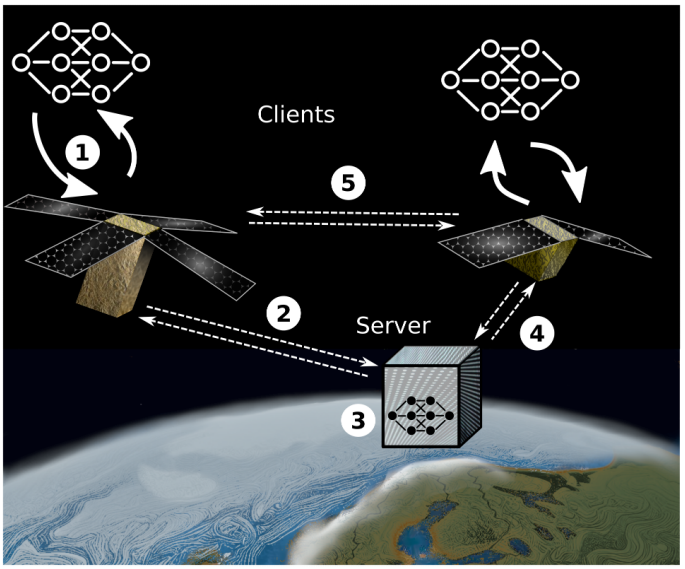


Figure 6.1: In Federated Learning, each satellite performs on-board machine learning to train a local model (1). Only these models are transmitted via satellite link to a server (2), here based on the ground, where multiple local models are aggregated into a single global model (3). This global model is transmitted back to the satellites (4) to continue the learning process. If necessary, satellites can act as relays for one another (5).

In Section 6.1, we first offer an initial exploration of the conceptual challenges associated

with this representative use case. We identify the characteristics of the problem, present a brief survey of the state of the art for each, connecting existing research from the application domain and the theoretical field, then discuss how existing approaches might fare in this scenario. We conclude with a gap analysis. Section 6.2 is dedicated to exploring how standardisation could support such use cases – another aspect critical to the deployment of novel technologies in the space domain. We examine the problem in two steps, beginning with the simple transmission of a pre-trained machine learning model from ground to satellite, then considering the repeated exchange of models as required by a federated learning algorithm.

6.1 | State of the art and application challenges

With the proliferation of private and commercial missions, an ever-increasing number of satellites with different capabilities and overlapping interests are active in Earth orbit. Enabling an ad-hoc collaboration between satellites of providers with compatible interests could serve to enhance the performance of all sides at a comparatively low communication cost. In this section, we analyse the current state of the art in research relating to this use case of cross-provider FL. As this use case has not yet been addressed explicitly, we divide our analysis into different thematic sections. We begin by considering the research closest to application, considering orbital edge computing (OEC) and Federated learning (FL) schemes tailored to use on satellites. Following this, we discuss cross-provider FL, an OEC use case that has, to the best of our knowledge, not been addressed to date. We explore related research from the field of FL that could be applicable for this use case, particularly works addressing the handling of different types of heterogeneity, and discuss their applicability.

6.1.1 | Orbital edge computing and federated learning

Federated Learning could offer a flexible framework for satellites to collaborate on on-board information processing[Izz22][Che22] while limiting communication cost and preserving data privacy. Various works modify the FL paradigm for the use case of LEO constellations, mainly focusing on adapted communication schedules to handle the intermittent connectivity of satellites. These approaches can broadly be divided by their proposed placement of a parameter server.

Initial works focused on the use of a ground-based server, offering a higher resource capacity than a satellite performing the same function; the drawback is a communication bottleneck caused by the intermittent connectivity of satellites. The FedSpace algorithm [So22] attempts to overcome this constraint by performing semi-asynchronous federated aggregation, exploiting knowledge about clients' orbital periods to calculate an aggregation schedule that yields an optimal trade-off between satellite idleness and model staleness. The FedGSM algorithm[Wu23] similarly makes use of known connectivity intervals to extrapolate model updates.

Conversely, FedISL[Raz22], a synchronous FL scheme for a dense LEO constellation, hinges on the strategic placement of a server in medium Earth orbit (MEO); with convergence speed further enhanced by the use of intra-plane inter-satellite links. This concept is extended in [Raz24] with the grouping of satellites sharing the same orbit to speed up aggre-

gation. Similarly, the synchronous FedHAP [Elm22b] algorithm is based on the deployment of multiple high-altitude aerial platforms to accelerate aggregation; AsyncFLEO [Elm22a] rests on the same premise, but is capable of asynchronous aggregation. The DSFL[Wu22] algorithm side-steps the challenge of server placement by performing fully decentralised aggregation. Finally, in their work on semi-supervised FL, Östman et al. [Öst23] compare a decentralised aggregation strategy with two scenarios where a relay satellite and a set of ground stations, respectively, act as the aggregation server. All three variants are shown to achieve comparable accuracy performances with similar total training time and power consumption.

Note that none of the works presented in this section consider heterogeneity challenges in great depth; several works, e.g. [So22][Raz24], claim that any suitable FL algorithm could be utilised as a drop-in component. In the following section, we assess the additional characteristics that might be required of an algorithm to mitigate heterogeneity in the cross-provider use case.

6.1.2 | Heterogeneity challenges of cross-provider FL

Satellite communication problems represent a natural example of a use case with heterogeneous participants. Machine learning has the potential to assist with various SatComm-related problems[Fou21], and collaboration between satellites could help solve these problems with greater accuracy and reliability. Many such satellites by different service providers are in operation today, with different hardware, different orbits and different underlying purposes, but nevertheless carrying out related functions. Compared to performing FL on single-mission satellite constellations, this application scenario presents unique challenges induced by the heterogeneity of satellites. Various types of heterogeneity are known to present a challenge to FL algorithms[Kai21b]. These include *data heterogeneity*, *feature heterogeneity*, *device heterogeneity*, and *preference heterogeneity*. In this section, we discuss the state-of-the-art approaches for each of these types, highlighting how each has been addressed for the OEC use case and, where missing, how existing solutions might transfer to this use case.

Data heterogeneity. This type of heterogeneity, where data is imbalanced across participants, is discussed extensively in the literature[Kai21a], as it occurs naturally in most real-world settings. In our use case, heterogeneous distributions of data across satellites are quite likely, with the extent dependent in part on the precise setting. For example, satellites gathering Earth observation images might collect significantly different samples based on their orbital planes, while for SatComm data might differ based on the role of the satellite or the associated service provider. The general issue of data heterogeneity is discussed in most FL variants proposed for the OEC use case, e.g. [Wu22], [Wu23]; however, their effectiveness is seldom demonstrated beyond preliminary benchmarking experiments. Therefore, it appears worthwhile to also consider the state of the art in the general field of FL. A taxonomy of variants of data heterogeneity is presented in [Ye23b], along with a comprehensive survey of current mitigation approaches. According to [Ye23b], these can be broadly divided into data-level, model-level and server-level interventions. Data-level approaches involve modifying the underlying training data to balance heterogeneity, e.g. by preprocessing [Li21a][Xu23], generating supplemental data using

Generative Adversarial Networks [Goo20], or transmitting information about data between clients [Yoo21]. However, these strategies often place a significant additional computing or communication burden on the clients, rendering them unattractive use on satellites. Selected model- and server-level strategies appear more promising, as they either require little additional computation cost, or can be carried out on the server-side. Notably, these include model regularisation [Kim22], knowledge distillation [Zhu21b], and personalised federated learning (PFL) approaches [Tan23] such as client clustering[Gho22a][Dua21a], parameter decoupling [Ari19] and model interpolation [Han20].

It is difficult to single out an optimal approach for the general version of our use case, where the data distribution pattern is unknown. The most promising approach would likely be an adaptive solution that modifies the aggregation approach during runtime based on observed metrics, e.g. clustering participants by similarity [Gho22b][Dua21b] or assigning importance weights for aggregation [Han20]. For mission architectures involving a powerful ground-based parameter server, more complex knowledge distillation-based approaches may also be an option.

Finally, we note that if the nature of the data distribution is known, such as in Earth observation imaging missions, this could be exploited to the advantage of the algorithm, e.g. by grouping participants known to collect similar data, or conversely by exchanging small sets of selected samples to balance highly different datasets.

Device heterogeneity. Aside from the communication challenges induced by orbital trajectories, satellites in the cross-provider setting would alsohave hardware differences, leading to different levels of sensor noise and training datasets of varying quality, and impacting computational speeds and capabilities. This is a common problem in the general field of FL [Ye23b]; standard approaches include adaptively assigning different weight contributions [Ma22] or model architectures [Dia20] to participants; possibly also reducing the consideration of lower-quality participants in selecting clients for aggregation [Li21b][Nis19]. It appears likely that such strategies would transfer well to the present use case, without a need for major modifications.

Feature heterogeneity. This setting corresponds to the collaboration of satellites with different types of sensors, collecting different types of data and potentially requiring different model architectures for on-board processing. Effectively integrating different features and models into a coherent federated model training process presents a difficult problem; to the best of our knowledge, it has not yet been tackled in research on OEC. Indeed, the general problem of performing FL in such a setting, known as Vertical Federated Learning (VFL) [Yan19], remains largely unsolved beyond highly constrained artificial scenarios or costly compensation approaches [Liu24]. This present lack of solutions renders the application of FL to the real-world satellite use case infeasible. The only approach that appears viable at present would involve feature distillation on a ground-station parameter server – a computationally expensive solution that does not yield workable models for the federated participants, and so would benefit only the server-side [Liu24]. Beyond this niche variant, the currently most feasible approach would likely be to separate participants by features, eliminating this type of heterogeneity.

Preference heterogeneity. This is a novel type of heterogeneity that, as of now, has seen little recognition in the field of federated learning; yet it appears highly relevant

to the present use case. Preference heterogeneity arises when participants are solving problems with multiple objectives, e.g. minimising communication cost while also minimising connection latency in a satellite communication network. In such multi-objective problems, different optimal solutions are generally possible, representing different trade-offs between the individual objectives. In practice, some trade-offs may be more desirable than others, e.g. conserving energy by limiting communication for severely resource-constrained satellites, or minimising connection latency to boost service to certain geographical areas. This can be controlled by assigning importance weights to each objective when solving the learning problem. We call participants with different importance weights preference-heterogeneous.

At present, little research has been devoted to performing Federated Learning under preference heterogeneity; the closest related works consider federated multi-task learning [Smi18], where participants have fully separate objectives, and federated multi-objective learning [Yan23b][Har23b] without allowance for different preferences.

6.1.3 | Other considerations

6.1.3.1 | Fairness between participants

In addition to the technical considerations of the previous section, this cross-provider use case also differs from the single-provider variant in the assumptions made about participants' intentions. Satellites designed to collaborate with each other as part of a single mission can generally be assumed to act altruistically in federation, i.e. towards the benefit of the larger system. In the cross-provider setting, however, no such assumption should be made, as has been noted e.g. in [Raz24]. Instead, we assume that participants may act 'selfishly', valuing their own success over that of the federated system. This could for example occur if satellites contribute low-quality updates, leading to an overall degradation of the global model and benefiting from the collaboration at the cost of others. Similarly, a participant could limit the frequency of its contributions to conserve communication budget or maintain privacy, to the detriment of others in the federation, while still receiving global model updates. A successful cross-provider collaboration scheme should guard against such exploitation. These fairness considerations have yet to be addressed in works targeting the OEC use case; a full survey on the state of the art of general FL approaches, not focused on this use case, is provided in [Shi24]. The same work gives a detailed account of the different definitions of fairness and underlying assumptions; the choice of an appropriate mitigation approach for the present depends on these characteristics.

6.1.3.2 | Protecting against malicious participants

This is a more extreme case of the challenges discussed in the previous section – here participants intentionally attempt to sabotage the performance of the federated system through their participation, e.g. through submitting intentionally false model updates (known as model poisoning attacks). A thorough overview of possible attack vectors and approaches for guarding against such attacks is given in [Kai21a] and more recently [Rod23], suggesting e.g. the assignment of confidence scores [Muñ19], filtering outliers [Yin18], or normalising model updates before aggregation [Sun19]. A number of these solutions appear to transfer well to the cross-provider OEC use case, given a trustworthy server, with the selection of approach dependent on the particular parameters of the system and the results

of a risk analysis.

6.1.3.3 | Standardisation

Along with the algorithmic approaches towards preventing misuse of the collaboration, standardisation likely has an important role to play in the deployment of FL to the present use case involving multiple providers. This could for example include a requirement for certification of machine learning pipelines in accordance with certain quality standards to obtain access to such federated exchange schemes, to decrease the likelihood of interference by malicious or poorly engineered participants. A review of standards relating to the trustworthiness of machine learning for space applications suggests that such standards largely have yet to be defined [Rei23a]; a recently published handbook provides a first glimpse of such considerations [ECS23].

Finally, we note another crucial challenge of this particular use case: the need for a unified communication protocol between participants, capable of negotiating the parameters of the FL scheme and able to transmit machine learning models unambiguously, without interpretation errors caused by differences in hardware or software. For ad-hoc collaboration, standardised communication formats are of critical importance, both to negotiate the parameters of the FL protocol during the initialisation phase, and to transmit model updates without error. There appears to be no existing solution, as most works in the literature tend to assume a group of satellites collaborating as part of a single unified mission. In the remainder of this chapter, we discuss this standardisation gap in more detail, and offer an initial exploration of how existing standards may serve as a basis for developing standardisation that supports the deployment of machine learning models in such contexts.

6.2 | Standardising space communication protocols for federated learning

This section is dedicated to examining how standardisation could support the communication aspect of deploying novel machine learning systems, such as Federated Learning, onto satellites. Standards should enable a stable transfer of models from ground to satellite and between satellites and ensure compatibility with existing space communication protocols. We focus our analysis on two broad use cases: In a first step, we consider the general problem of transmitting a single complete machine learning model from a ground station to a satellite. This corresponds to a scenario, illustrated in Fig. 6.2, where the training of a machine learning model is carried out on the ground, using previously collected raw sensor data from the satellites or a synthetic approximation as the underlying training data. The trained model is then deployed from the ground station onto the satellite. The capability to successfully carry out this procedure deploying and running a machine learning model on a single spacecraft is a fundamental stepping stone on the path towards the integration of more complex machine learning pipelines into space missions.

In the second part of our analysis, we consider one such more complex scenario building on the first: enabling on-board machine learning in distribution across multiple satellites, using the Federated Learning (FL) paradigm. Under this paradigm, each satellite trains an

on-board machine learning model, periodically exchanging information about the resulting models with other satellites to enhance the learning process – see Fig. 6.3 for an illustration. This scenario necessitates an extension of the communication standard required for the first case.

6.2.1 | The role of standardisation

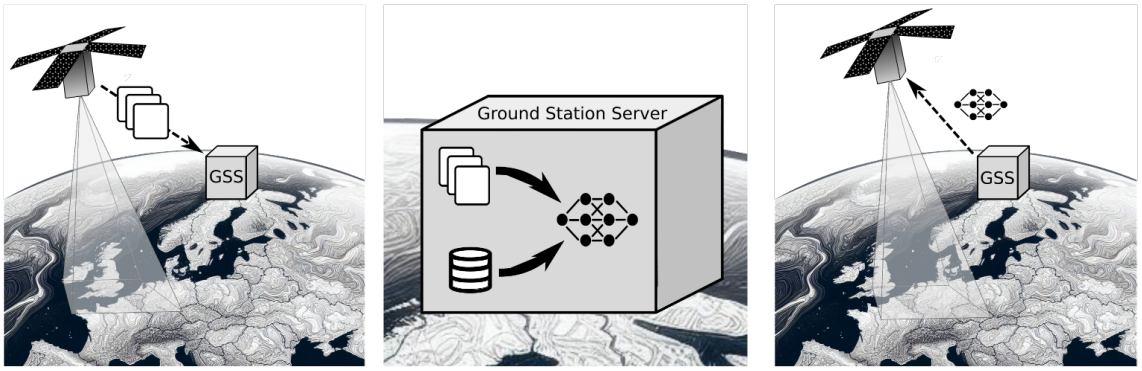
Standardisation is an important tool to codify insights and advances in technical research, building a solid foundation for further progress and preventing errors in the implementation of novel technologies. This is of immense importance in the aerospace domain, and particularly so for spaceflight missions, where failure is punished swiftly and harshly by the uncompromising environment of space. Clear standards could help in every part of the development pipeline: to ensure compliance and interoperability during the design phase, to establish rigorous and clear expectations for performance, permit later integration of independently developed components, allow correct performance during deployment and systematic troubleshooting in case errors do occur.

Beyond ensuring the effective deployment of existing technologies, standardisation is also a crucial tool to assist in the safe exploitation of new technologies, such as Artificial Intelligence and Machine Learning in the spaceflight domain. However, the standardisation of machine learning presents a particular challenge, as the state of the art in the domain is progressing at such a swift pace that it is difficult to identify and formalise enduring characteristics of the technology. Compared to traditional systems, research on machine learning is evolving rapidly, requiring standards that are both flexible and robust. At the same time, these new developments in machine learning are adopted at a much more accelerated rate than technological changes have historically been embraced by the industry, further increasing the urgency of standardisation development.

To support the utilisation of machine learning models on-board spacecraft, standards for the development and deployment of such models are required, including qualifying and quantifying the trustworthiness characteristics of systems [Rei23b]. Once the quality and trustworthiness of a model can be established, the next great challenge is the cross-device communication of such a model in practice. In the aerospace domain in particular, standardising communications protocols is of crucial importance: without well-defined communications protocols, all other aspects of a spaceflight mission are at risk. Standards can prevent known problems and even enhance the flexibility of a mission, allowing interactions that were not necessarily planned during the design phase of the mission, enabling compatibility between independently developed systems. With the current drive towards multi-satellite missions and the diversification of stakeholders, this aspect gains importance, as it becomes more feasible and desirable for different satellites to collaborate.

6.2.2 | First case study: ground-to-satellite model transfer

In this case study, we explore how a machine learning model may be deployed from the ground onto a satellite, outlining concrete approaches taken in existing missions and discussing how these relate to current standardisation efforts. Given the subject of this work, our discussion will mainly focus on the network aspects of potential solutions, i.e. how a machine learning model can be encoded for transmission, and how the resulting information



(a) A satellite collects data during operations and sends them to a ground-based server.

(b) The centralised training data is used by a ground server to train a machine learning model.

(c) The completed model is deployed onto the satellite.

Figure 6.2: The scenario of the first case study. We consider how to facilitate the communication of the machine learning model from the ground to the satellite, taking place in the last step.

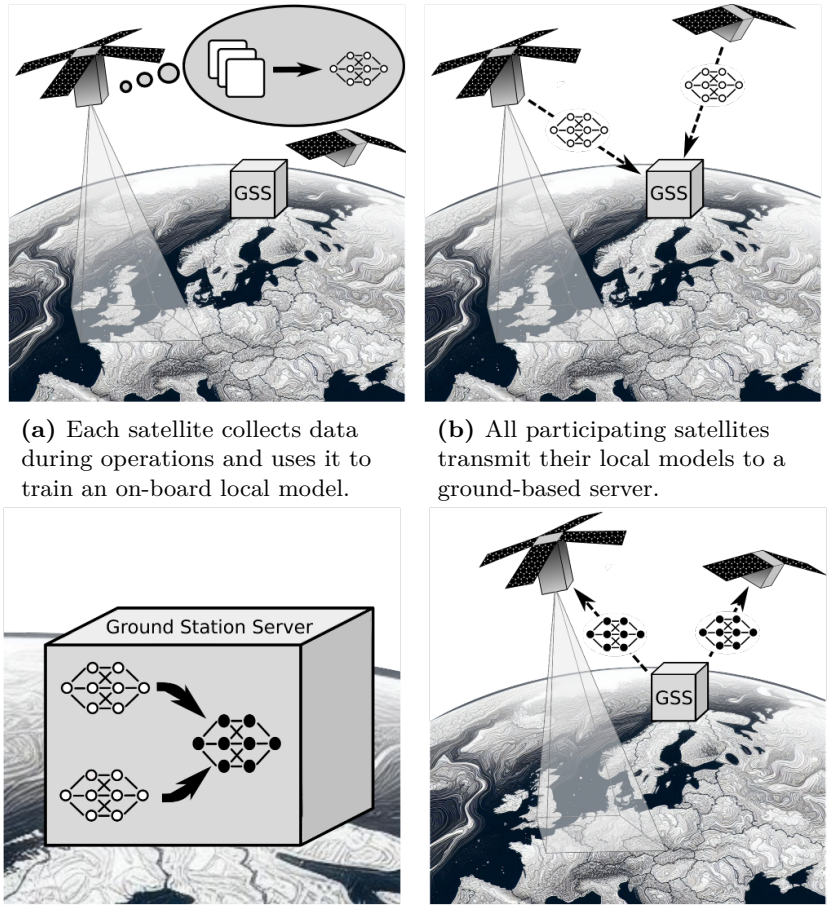
is transmitted.

The deployment of machine learning models on-board spacecraft is still in its initial stages; we therefore approach our analysis from two sides. We begin by considering existing standards and methods for encoding and communicating machine learning models in the general case. Then, we discuss the solutions chosen for the Φ-sat-1 (PhiSat-1) mission [Giu22], a recent proof-of-concept mission where a machine learning model was successfully deployed onto a satellite. Finally, we examine how a model transfer procedure could be integrated into existing space communication standards formulated by the Consultative Committee for Space Data Systems (CCSDS).

6.2.2.1 | Model transfer formats

Historically, machine learning models have been developed, trained, and deployed in fairly self-contained pipelines, each designed for a specific purpose and for the needs of a specific stakeholder. In practice, this has caused different programming frameworks fit for different purposes to proliferate, with no particular requirement for mutual compatibility. Not only do these frameworks encode and store models in different formats; they also translate to different behaviours upon hardware deployment. In effect, this means that models containing apparently the same abstract structures might deliver substantially different results when deployed by different frameworks.

With the recent move towards collaborative machine learning strategies and the increasing interest in deploying machine learning models across edge devices, the need for aligning different frameworks has become apparent. The absence of a unified standard introduces risks such as interoperability issues, increased development costs, and the potential for mission-critical failures. Given the entrenched differences in machine learning frameworks, the most straightforward way of facilitating alignment would be to establish a unified model transfer format, designed to allow models to be communicated unambiguously between



(a) Each satellite collects data during operations and uses it to train an on-board local model.

(b) All participating satellites transmit their local models to a ground-based server.

(c) The local models are aggregated into a global model by the server.

(d) The new global model is deployed onto the satellites.

Figure 6.3: The scenario of the second case study. We consider how to facilitate the communication of the machine learning model between ground and satellites, taking place throughout the training process.

frameworks.

To date, no such unified model transfer format has been defined by an independent standardisation body. Outside of the formal standardisation domain, two main competing model transfer formats currently exist: the Neural Network Exchange Format (NNEF, [Gro25]), and the Open Neural Network Exchange format (ONNX, [com25]). Neither of the two formats can be considered an officially recognized standard as defined by Regulation (EU) No 1025/2012[Cou12], since both are being maintained by different industrial stakeholder consortia. Several additional tools of more limited scope exist, designed either to support specific providers, such as Intel's OpenVino tool[Int25], or to convert between specific frameworks on the application level.

For this study, we are most interested in a general solution that is applicable across a

variety of hardware configurations; hence we place our focus for this work on the capabilities of the NNEF and ONNX formats. In particular, we consider how such a general solution could be integrated into the existing framework of related standardisation in the relevant domains.

Closer investigation reveals that the ONNX format relies on the serialisation of the computational graph, including parameter values, that corresponds to a neural network into a binary format for transfer. In contrast, the Neural Network Exchange Format, maintained by the Khronos NNEF Working Group, consists of a description of the complete computational graph corresponding to the structure of the given neural network, expressed in a human-readable syntax, and the numerical parameters associated with the network. A number of common machine learning tools support the import and export of models in NNEF format. Of the two formats, NNEF appears to be preferred by domain experts for potential application to the aerospace domain, as seen e.g. in [Gau23], due to its relatively well-documented syntax and semantics.

6.2.2.2 | Existing solutions: PhiSat-1

Particularly of note for inspiration from the application domain is the PhiSat-1 (and PhiSat-2) mission, launched as a proof-of-concept mission by the European Space Agency (ESA). The purpose of this mission was to provide a first technology demonstration of running Artificial Intelligence methods for Earth observation on-board a nanosatellite, including the development and deployment pipeline, hardware capability, and analysis of the results. To date, this is the only European mission to demonstrate the deployment of a pre-trained artificial Intelligence model for Earth observation onto a satellite in space. This mission design matches our scenario; hence it is useful to consider how it was realised – both from a standardisation perspective, and as a demonstration that the technology is feasible or will soon be ready for deployment, making the timing of these standardisation efforts crucial.

The PhiSat-1 mission makes use of the proprietary OpenVino tool, developed specifically for Intel hardware, to facilitate the transfer of the pre-trained model onto the satellite. This choice does not generalise well in terms of standardisation, as it does not cover hardware components built by other providers. From a standardisation perspective, we are in contrast interested in building a more universal, hardware-agnostic standard solution.

Beyond the model transfer format, we note that the general communication with the satellite and software deployment was facilitated using the Nanosatellite MO Framework (NMF), which implements common CCSDS standards to facilitate communication [Coe17]. This could be of interest for further investigation into the pairing of a general model transfer format with common CCSDS standards, and the network stack could perhaps even serve as another instantiation example in addition to our case study.

6.2.2.3 | Integration with CCSDS: proposed communications stack

The Consultative Committee for Space Data Systems (CCSDS) defines a large number of communication specifications for use in space applications. Some CCSDS standards have been adopted as European standards within the CEN/CLC JTC 5 committee and as international standards within ISO/TC 20/SC 13. Given their widespread use, it is

worthwhile to consider if and how the domain-agnostic model transfer formats discussed in the previous section could be integrated into this protocol stack.

For the present use case, this appears to be quite uncomplicated: CCSDS protocols are classified following the standard Open Systems Interconnection (OSI) networking model[Int94]. The transfer of a machine learning model from ground to a satellite using a model transfer format, as discussed in this chapter, could likely be encapsulated in the application layer, using a communication stack of existing CCSDS protocols. A possible instance of such a network stack is proposed in Table 6.1, with brief citations of the related CCSDS standards to support the respective choices. This table is primarily intended to serve as an illustration; the specific choice of instantiation should be dependent on mission configuration.

Table 6.1: Example instantiation of a network stack that may be used to transmit encoded machine learning models to satellites.

OSI Layer	Example Protocol	Comment
Application	CFDP + Lossless Data Compression	"CFDP is designed to meet the needs of space missions to transfer files. It is a file transfer protocol, but it also provides services typically found in the Transport Layer, that is, complete, in-order, and without duplicate data delivery."[Spa23, p. 3-10] The Lossless Data Compression standard "guarantees full reconstruction of the original data without incurring any distortion in the process. It is intended to be used together with the Space Packet Protocol or CFDP."[Spa23, p. 3-11]
Transport	SCPS-TP	
Network	Bundle Protocol (BP)	
Data Link	Unified Space Link Protocol (USLP)	USLP has "a function for retransmitting lost or corrupted data to ensure delivery of data in sequence without gaps or duplication over a space link." [Spa23, p. 3-2]
Physical	CCSDS Recommended Standard for Radio Frequency and Modulation Systems	See [Spa21].

In the next section, we will discuss a more complex use case building on the scenario discussed here.

6.2.3 | Second case study: Federated Learning across satellites

In this section, we discuss the extension of communication protocols required to enable collaborative on-board machine learning across satellites. The small size and limited resources of modern nanosatellites require the collaboration of multiple satellites to enhance on-board computing capabilities, allowing resources and information to be shared across spacecraft.

6.2.3.1 | Federated Learning Protocol

Given the wide variety of existing Federated Learning algorithms, any standardised deployment of FL would need to begin with a clear and unambiguous communication of the specific characteristics of the algorithm being deployed. The exchange of models, which can exploit one of the formats discussed in Section 6.2.2, is only one of the aspects needed for the successful implementation of an FL communication scheme. We suggest the development of a dedicated Federated Learning protocol to set up and facilitate any ad-hoc collaboration between multiple satellites under the Federated Learning paradigm. No such protocol currently exists. In this section, we provide a first overview of the challenges to be solved by such a communication protocol by suggesting a list of characteristics that it should address. This list is intended to serve as a starting point; it is by no means exhaustive. Our proposals for parameters that should be included in this protocol are split into two subsets: the set of client parameters, shown in Table 6.2, which define the behaviour of participants in the client role of the federated learning system, and the set of server parameters, shown in Table 6.3, which specify the behaviour of the federated server.

6.2.3.2 | Communication of model updates

Once the initial parameters of the federated learning scheme have been established, models must be transmitted periodically to and from the clients during the training process. In this section, we consider how these model transfer messages might be realised.

The simplest solution in terms of standardisation effort would be to transmit the complete model for each update, using a model transfer format as established in the previous section. However, this would involve a significant needless expenditure of energy, as most common federated learning schemes do not modify the underlying architecture of the machine learning model during the learning process. If the underlying model structure remains fixed, the format of model updates could then be reduced to updating only those aspects of the model that do change, i.e. the scalar weights assigned to the nodes of a neural network. As the reduction of communication cost is a crucial challenge in the design of energy-efficient space missions, this possibility bears further consideration.

One of the two existing model transfer formats considered in this chapter, the NNEF format, appears to be suited to this strategy with little modification required. In this format, model information is encoded in human-understandable semantics, with information about the model architecture and the parameter weights assigned to this model stored separately. Retrieving and transferring only the parameter-weight section of the encoded format, and integrating this partial update into the model on the receiving end, should present little additional difficulty.

On the other hand, implementing the same strategy with the ONNX transfer format

appears to be much more complex, as this format encodes all model information in a single binary file, with no obvious way of isolating the model parameters. Doing so would likely require more significant modifications to the underlying encoding of the format than appears practical.

Table 6.2: Proposed client parameter information that a federated learning communication protocol should encode.

Name of client parameter	Description	Comments
System topology	Different federated algorithms use different aggregation control schemes, e.g. fixed star topology (one central server), fully decentralised (no server at all), etc.	
Server identity	Identity/address of the server, if it exists	Needs to be fixed for clients if (1) clients are required to contact server, or (2) for security, to allow clients to verify server if contacted
Local submission trigger	Defines how model submission is triggered on the local client: after a given number of steps, by reaching a certain training loss, by the server, etc.	
Local aggregation behaviour	Defines client behaviour after local model has been submitted, but before external model update has been received. Behaviours could include pausing training until model update is received, or continuing training.	
Local integration of global update	Defines how a global model update is processed on the local client, e.g. replacing the local model, partial update, etc.	
Local submission format	Defines the type and format of local updates to be submitted by the client. E.g. weights of full model, gradient of partial model, etc.	
Initial model	Defines the architecture (or fully initialised version) of the model to be trained on the client.	Can be reduced to a set of constraints if the FL scheme does not require homogeneous client models.
Failure handling	Defines how to detect and handle different failures locally, e.g. time-out on global updates if server fails, procedure for handling server changes or local failures	Requires detailed specification of additional fields

Table 6.3: Proposed server parameter information that a federated learning communication protocol should encode.

Name of server parameter	Description	Comments
Global aggregation behaviour	Aggregation strategy to be used on the server	
Model collection mechanism	Defines how models are collected from the local clients, e.g. through active collection by the server or proactive client submission	
Initial (global) model	Defines the expected architecture of the global model.	Can be transmitted using model transfer format discussed in previous section.
Failure handling	Defines how to detect and handle different failures locally and in the system, e.g. time-out on local updates if client fails, procedure for handling server changes or local failures	Requires detailed specification of additional fields

6.3 | Summary

In our analysis of the state of the art in Orbital Edge Computing, we have seen that current works investigating the use of Federated Learning in satellite on-board edge computing have largely focused on a single scenario: a large constellation of homogeneous satellites deployed as part of a single mission. We have introduced a different use case, relevant to the present or near future, where heterogeneous satellites of multiple different providers could collaborate under a FL scheme to enhance on-board learning. We have elucidated the unique conceptual challenges of this use case, with a focus on the different types of heterogeneity and conflicts of interest that might arise.

In Section 6.1, we have provided a broad perspective of the state of the art for each, discussing both the existing work close to the use case, and the general state of the art in the theoretical literature. Our brief survey shows that several aspects remain to be addressed to adequately solve this real-world use case. In particular, there is a need to further investigate (1) how state-of-the-art solutions can be combined in settings where multiple types of heterogeneity occur simultaneously; (2) which heterogeneity-mitigating algorithms could be selected to fit with the use case, and with existing OEC schemes; (3) how to perform Federated Learning under preference heterogeneity. Finally, we suggest that any engineering approach should be supported by additional measures reducing the complexity of the system by providing appropriate constraints, e.g. through the use of standardisation.

In Section 6.2, we have discussed in more detail how such standards may be constructed: In a first case study in Section 6.2.2, we have identified a standardisation gap covering communication protocols for the transmission of machine learning models between ground and spacecraft, and spacecraft-to-spacecraft. We have discussed existing standardisation and industry standards addressing the transfer of models in other domain contexts, and how these existing formats could be encapsulated in existing space communication standards. Finally, the second case study, discussed in Section 6.2.3, identified a need for the development of a dedicated communication protocol to facilitate a future use case where multiple satellites cooperate in training a machine learning model using a Federated Learning strategy. Suggestions for tackling this standardisation gap in an effective manner were proposed, including a list of characteristics to consider in defining such a communication protocol, and how to leverage existing related standards to optimise the size of communication messages transmitted during the learning process.

7 | Conclusion and Perspectives

Contents

This thesis has considered the integration of Federated Learning with multi-objective methods, with a focus on laying the foundation for the hereto largely unexplored direction of Federated Multi-objective Learning. Effective FMOL strategies could help leverage the twin advantages of Federated Learning, which facilitates machine learning in distributed settings where collaboration would not otherwise be possible, and multi-objective methods, which improve the applicability of optimisation methods by reflecting the complexities of the real world.

7.1 | Summary

In Chapter 2, we have discussed the state of the art and introduced a first taxonomy to classify existing works that combine Federated Learning and Multi-Objective Optimisation methods. This taxonomy offers for the first time comprehensive, clear, and unique labels for different methods in the field, ensuring clarity in future discourse and contributions. Chapters 3 and 4 contain algorithmic contributions that fill gaps in FMOL: Chapter 3 presents a first framework to formalise Pareto-based Federated Multi-objective Learning, while Chapter 4 proposes FedPref, an algorithm designed to address a novel setting

where distributed participants solve the same multi-objective problem under heterogeneous preferences. Chapter 5 tackles the problem of benchmarking FMOL algorithms, arguing that current benchmarks are not sufficiently representative of the true difficulty of the problem space, and proposing a more difficult class of benchmarking problems based on the domain of Fair Machine Learning. Finally, Chapter 6 offers a view on potential future applications of the FL approach to the aerospace domain.

7.1.1 | Q1: What are the commonalities, differences and challenges in combining multi-objective methods with federated learning?

We have shown that contributions can be broadly separated into three categories based on the level of integration of multi-objective techniques in the federated system: top-level integration for off-line hyperparameter selection, federation-level integration for on-line control of the behaviour of the federated system as a whole, and client-level integration to solve arbitrary multi-objective problems in federation. The level of integration has fundamental consequences not only for its different use cases, but also for the extent and type of modification that is necessary to the federated algorithm. Building on this insight, we have proposed a comprehensive taxonomy, surveying and classifying existing works in the literature and identifying gaps open for future study.

7.1.2 | Q2: How can multi-objective learning problems be solved in federation?

We have tackled two different federated multi-objective settings in this work: one where clients with no fixed preferences collaborate to find a Pareto front of solutions, and another where clients have individual fixed preferences that are not shared with the server. For the first scenario, we have proposed a general framework, the first to address this setting, to solve such problems using decomposition. We have described the general framework and validated it using a representative instantiation on a series of experimental configurations, using a set of multi-objective reinforcement learning problems.
For the second scenario, we have identified preference heterogeneity between clients as a novel, previously unaddressed challenge in Federated Learning. We have proposed a Personalised Federated Learning algorithm, dubbed FedPref, to collaboratively train individual models tailored to the specific preferences of each client. This algorithm relies on similarity-based clustering and weighted aggregation, making it implicitly compatible with other types of heterogeneity as well. We have evaluated the performance of the algorithm extensively, showing across five different MORL benchmarking environments and three different random preference distributions that this algorithm reliably matches or outperforms other state-of-the-art heterogeneity-mitigating FL algorithms. Following this validation of the client performance, we have extended our evaluation to the system level, introducing a multi-objective view of the federation as a whole using classical MOO metric. Evaluating the diversity and convergence of federated solutions in this way permits additional insight into the ability of the federated algorithm to honour diverse client preferences.

7.1.3 | Q3: How can federated multi-objective learning algorithms be validated in a general way?

To answer this question, we have begun by considering the existing approach of transferring multi-task benchmarks from the centralised setting. We have argued, and shown a supporting example, that this class of problems does not generally contain inherent conflict between the tasks used as individual objectives. As such, these problems do not adequately represent the main challenge of multi-objective problems. To rectify this issue, we have proposed a framework for constructing an additional class of benchmarks, derived from the field of fair machine learning, that permits the use of a number of objectives known to conflict. Numerous existing benchmarking datasets from the field of fair machine learning are compatible with this approach. This benchmarking approach is lightweight and can be used as a drop-in extension of existing FMOL implementations, requiring only simple additions. We have demonstrated the use of this new class of benchmarks by implementing ten different instantiations and setting a number of state-of-the-art algorithms to solve them. The results show that these benchmarks do present a challenge to existing algorithms, with different algorithms exploring different sections of the Pareto front.

7.1.4 | Q4: What other challenges currently hinder the application of FL methods in complex real-world use cases, such as the space domain?

We have addressed this question by first identifying a plausible near-future use case for Federated Learning methods in the space domain, taking into account currently existing mission configurations and observed trends in the field. Based on these considerations, we have analysed the challenges remaining in principle for the deployment of Federated Learning on heterogeneous satellites in Earth orbit. We have noted that work remains to be done to bridge the gap between theory and practical application of Federated Learning. Of particular interest to the space domain are methods that can handle various types of heterogeneity and robustness guarantees.

Another crucial aspect, currently lacking, is the development of standards to support the deployment of such methods. We have analysed two case study scenarios, one on the transfer of a trained machine learning model from a ground station onto a satellite - close to the current technological state of nanosatellite missions, such as the PhiSat-1 mission - and one representing a more ambitious near-future use case where multiple nanosatellites collaborate to perform on-board machine learning. For the first case study, we observe that no current dedicated standard exists, but that some existing standards for transferring machine learning models, developed in the general machine learning community, could likely be combined with comparatively little additional effort to fit with established CCSDS communication standards. For the second scenario, we have identified an additional standardization gap to be overcome: the need for a well-defined protocol allowing ground- and space-based participants to negotiate their participation in a federated learning system and establish the joint parameters of this collaboration. Our analysis has yielded a broad overview of items of interest that would need to be included in the definition of such a protocol to cover a reasonable range of existing FL approaches. Finally, we have discussed how our suggested solution for the first use case could be integrated into this second

scenario, and how messages defined in compliance with the NNEF model transfer format and passed during the main Federated Learning phase could be reduced effectively to conserve communication resources.

Ultimately, we have identified an achievable path towards the establishment of communication standards enabling the transfer of machine learning models in a space context in compliance with existing standardization, and demonstrated the feasibility of extending such standardization further to encourage the implementation of innovative distributed learning solutions.

7.2 | Limitations and future research

As the field of federated multi-objective learning is still in its infancy, it remains wide open to further research. In this section, we briefly discuss the limitations of our work, and indicate potential avenues for further research.

MOFL/D: A framework for federated multi-objective learning

To the best of our knowledge, this work was the first to consider the general case of multi-solution federated multi-objective learning and present a systematic approach to solving it using decomposition. As such, it remains open to further improvement. In particular, the current approach of using a full federated system with homogeneous client preferences may not be an optimal use of computing resources. Integrating a preference-heterogeneous multi-solution FMOL algorithm with the framework could allow for a more efficient exploration of the candidate space. In addition, this initial work did not explicitly take into account the potential for data or device heterogeneity between clients. These complications could presumably be mitigated by either instantiating the framework with federated algorithms designed to mitigate the given type of heterogeneity, or possibly by modifying the behaviour of the optimisation layer, including the candidate-generating heuristic or the client sampling procedure. Finally, potential further work includes investigating other, more complex possible instantiations and application to different types of multi-objective problems.

7.2.1 | FedPref: personalised federated multi-objective learning under preference heterogeneity

As this work presents a very first solution tailored to the objective-heterogeneous setting, several challenges inherent to the federated setting remain to be addressed in future work. This includes, in particular, scenarios dealing with combined occurrences of different types of heterogeneity, such as data or hardware heterogeneity combined with the preference heterogeneity discussed here. In principle, we expect that the model similarity-based design of our algorithm could adapt without change to a setting that includes data heterogeneity; heterogeneity induced by differences in client capabilities might require the integration of additional strategies dedicated to this purpose. Such strategies already exist in the literature; their integration into the cluster-aggregation step of FedPref appears quite feasible.

In addition, we have introduced the multi-objective view of the federated system in this work. Our experiments demonstrate that it appears to be possible to design a personalised federated algorithm that achieves both high individual client performance and a diverse set of client solutions for different preferences. Nevertheless, further study of the implications of various MOO metrics in this setting would be useful. Furthermore, it would be interesting to apply this multi-objective analysis to other federated algorithms from the state of the art.

7.2.2 | A new class of benchmarks for federated multi-objective learning

Fairness datasets are often highly imbalanced with respect to sensitive and classification labels. In a federated context, this introduces two issues. First, non-converged models may return deceptively high accuracy and fairness values – in most fairness metrics, blanket predictions of a fixed value are by definition perfectly fair. Such results should be excluded in analyses and parameter searches – we describe a simple filtering strategy in the appendix. Secondly, some variability in client-level results, particularly under heterogeneous preferences, likely comes from naive random partitioning of data. Future benchmarks would benefit from splitting strategies that preserve the distribution of all pairs of labels and sensitive attributes across clients.

7.2.3 | Towards real-world application in the aerospace domain

As shown by our brief analysis in Chapter 6, work remains to be done for a robust, flexible, and secure deployment of federated learning on satellites. Key challenges that we have identified include compensating for different types of heterogeneity between participants, particularly heterogeneity not based on imbalanced data; ensuring fairness between selfish participants; and the challenge of protecting against malicious participants effectively without introducing a significant loss in performance.

Furthermore, the aerospace domain in particular relies fundamentally on standards to ensure the correct, safe, and trustworthy deployment of novel technologies. Standardisation of networking protocols for transmitting machine learning models and setting up collaborations between satellites is urgently required. In this work, we have suggested a potential pathway for the rapid introduction of such standards, but significant additional work is required to codify these.

7.3 | Contributions

This section contains a comprehensive list of contributions produced over the course of this thesis, separated by category and ordered by year. Peer-reviewed scientific publications are listed first, followed by contributions to standardisation-related publications, awards, and outreach work.

Peer-reviewed publications

1. **Hartmann, Maria**, Grégoire DANOY, Mohammed ALSWAITTI, and Pascal BOUVRY: 'A split-training approach to JoVe-FL'. *International Conference on Optimization and Learning*. Extended abstract. 2023.

2. **Hartmann, Maria**, Grégoire DANOY, Mohammed ALSWAITTI, and Pascal BOUVRY: 'Efficient On-board Learning for Distributed Mission Configurations'. *Artificial Intelligence Symposium on Theory, Application and Research (AISTAR)*. Extended abstract. 2023.
3. **Hartmann, Maria**, Grégoire Danoy, Mohammed Alswaitti, and Pascal Bouvry: 'JoVe-FL – A Joint-embedding Vertical Federated Learning Framework'. *International Conference on Agents and Artificial Intelligence*. 2023.
4. **Hartmann, Maria**, Grégoire Danoy, Mohammed Alswaitti, and Pascal Bouvry: 'MOFL/D: A Federated Multi-objective Learning Framework with Decomposition'. *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*. 2023.
5. **Hartmann, Maria**, Grégoire DANOY, and Pascal BOUVRY: 'Heterogeneity: An Open Challenge for Federated On-board Machine Learning'. *European Space Agency SPAICE Conference / IAA Conference on AI in and for Space*. 2024.
6. **Hartmann, Maria**, Grégoire DANOY, and Pascal BOUVRY: 'Introducing FedPref: Federated Learning Across Heterogeneous Multi-objective Preferences'. *Multi-Objective Decision Making Workshop at ECAI 2024*. 2024.
7. **Hartmann, Maria**, Grégoire DANOY, and Pascal BOUVRY: 'A New Class of Benchmarks for Federated Multi-objective Learning'. Under review at ICLR 2026. 2025.
8. **Hartmann, Maria**, Grégoire Danoy, and Pascal Bouvry: 'FedPref: Federated Learning Across Heterogeneous Multi-objective Preferences'. *ACM Trans. Model. Perform. Eval. Comput. Syst.* (May 2025), vol. 10(2).
9. **Hartmann, Maria**, Grégoire Danoy, and Pascal Bouvry: 'Multi-objective methods in Federated Learning: A survey and taxonomy'. *International Workshop on Federated Learning with Generative AI In Conjunction with IJCAI 2025 (FedGenAI-IJCAI'25)*. 2025.

Standardisation-related publications

1. Reiff, Jean-Marie, Jean-Philippe Humbert, Pascal Bouvry, Grégoire Danoy, Mohammed Alswaitti, Manuel Combarro Simón, **Maria Hartmann**, Hedieh Haddad, Lucas Cicero, Jean Lancrenon, Nicolas Domenjoud, Leslie Fouqueray, Natalia Vinogradova, and Ruddy Enguehard: *Trustworthiness in ICT, Aerospace and Construction Applications*. Institut luxembourgeois de la normalisation, de l'accréditation, de la sécurité et qualité des produits et services (ILNAS), 2023.

2. Humbert, Jean-Philippe, Pascal Bouvry, Grégoire Danoy, Mohammed Alswaitti, Manuel Combarro Simón, **Maria Hartmann**, Hedieh Haddad, Lucas Cicero, Jean Lancrenon, Natalia Vinogradova, and Victoria Mletzak: *Research-driven Standardization Opportunities for ICT, Construction and Aerospace*. Institut luxembourgeois de la normalisation, de l'accréditation, de la sécurité et qualité des produits et services (ILNAS), 2024.

Awards

1. *CEN/CENELEC Standards+Innovation Award - Young Researcher category*. Presented annually to an individual student, under 30 years of age, based on the work done for academic theses, doctoral dissertations or other university research project addressing standardisation, following nomination by a national European standardisation body. 2024.

Outreach

1. 'Swarms of Nano-satellites'. Talk at ILNAS Journée mondiale de la normalisation (World Standards Day). 2022.
2. 'Swarms of nano-satellites'. Talk at ILNAS workshop "Space and technical standardization". 2022.
3. 'Federated Learning for Swarms of Nano-Satellites'. Talk at Technoport. 2023.
4. 'Federated Multi-objective Learning with Decomposition'. Talk at University of Luxembourg. 2023.
5. 'Presentation of aerospace section of the white paper "Trustworthiness in ICT, aerospace and construction applications Scientific research and technical standardization"'. Talk at ILNAS Journée mondiale de la normalisation (World Standards Day). 2023.
6. 'Round table on cybersecurity in the aerospace domain'. Talk at ILNAS workshop "Technical Standardization in Space and Cybersecurity". 2023.
7. 'A survey of Personalised FL and Federated Multi-objective Learning'. Talk at 2nd Summer School on Federated Machine Learning. 2024.
8. 'Federated Multi-objective Learning with Heterogeneous Preferences'. Talk at University of Luxembourg. 2024.
9. 'Presentation of aerospace section of the white paper "Research-driven Standardization Opportunities for ICT, Construction and Aerospace"'. Talk at ILNAS Journée mondiale de la normalisation (World Standards Day). 2024.
10. 'An Introduction to Federated (Multi-objective) Learning'. Virtual talk at ACE Lab, Johns Hopkins University. 2025.

Bibliography

[Abe19] Abels, Axel, Diederik M. Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher: *Dynamic Weights in Multi-Objective Deep Reinforcement Learning*. May 13, 2019 (cit. on pp. 27, 47).

[Ale19] Alexandropoulos, Stamatios-Aggelos, Christos Aridas, Sotiris Kotsiantis, and Michael Vrahatis: 'Multi-Objective Evolutionary Optimization Algorithms for Machine Learning: A Recent Survey'. May 2019: pp. 35–55 (cit. on p. 22).

[Ang22] Angwin, Julia, Jeff Larson, Surya Mattu, and Lauren Kirchner: *Machine Bias*. ProPublica, 2022. Chap. 6 (cit. on pp. 71, 74).

[Ari19] Arivazhagan, Manoj Ghuhan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary: *Federated Learning with Personalization Layers*. 2019 (cit. on p. 86).

[Ask24] Askin, Baris, Pranay Sharma, Gauri Joshi, and Carlee Joe-Wong: *Federated Communication-Efficient Multi-Objective Optimization*. 2024 (cit. on pp. 18, 68, 73, 76, 137).

[Bad24] Badar, Maryam, Sandipan Sikdar, Wolfgang Nejdl, and Marco Fisichella: 'FairTrade: Achieving Pareto-Optimal Trade-Offs between Balanced Accuracy and Fairness in Federated Learning'. *Proceedings of the AAAI Conference on Artificial Intelligence* (Mar. 2024), vol. 38(10): pp. 10962–10970 (cit. on p. 17).

[Ban22] Banerjee, Sourasekhar, Xuan-Son Vu, and Monowar Bhuyan: 'Optimized and Adaptive Federated Learning for Straggler-Resilient Device Selection'. *2022 International Joint Conference on Neural Networks (IJCNN)*. 2022: pp. 1–9 (cit. on p. 17).

[Bec96] Becker, Barry and Ronny Kohavi: *Adult*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20. 1996 (cit. on p. 74).

[Bri20] Brik, Bouziane, Adlen Ksentini, and Maha Bouaziz: 'Federated Learning for UAVs-Enabled Wireless Networks: Use Cases, Challenges, and Open Problems'. *IEEE Access* (2020), vol. 8: pp. 53841–53849 (cit. on p. 3).

[Cai23] Cai, Ruisi, Xiaohan Chen, Shiwei Liu, Jayanth Srinivasa, Myungjin Lee, Ramana Kompella, and Zhangyang Wang: 'Many-Task Federated Learning: A New Problem Setting and A Simple Baseline'. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Vol. 34. Canada: IEEE, June 2023: pp. 5037–5045 (cit. on pp. 38, 42, 48, 127, 128).

[Cas22] Castelnovo, Alessandro, Riccardo Crupi, Greta Greco, Daniele Regoli, Ilaria Giuseppina Penco, and Andrea Claudio Cosentini: 'A clarification of the nuances in the fairness metrics landscape'. *Scientific reports* (2022), vol. 12(1): p. 4209 (cit. on p. 73).

[Cel19] Celis, L. Elisa, Lingxiao Huang, Vijay Keswani, and Nisheeth K. Vishnoi: 'Classification with Fairness Constraints: A Meta-Algorithm with Provable Guarantees'. *Proceedings of the Conference on Fairness, Accountability, and Transparency*. FAT* '19. Atlanta, GA, USA: Association for Computing Machinery, 2019: pp. 319–328 (cit. on p. 73).

[Cha22] Chai, Zheng-yi, Chuan-dong Yang, and Ya-lun Li: 'Communication efficiency optimization in federated learning based on multi-objective evolutionary algorithm'. *Evolutionary Intelligence* (Apr. 2022), vol. 16(3): pp. 1033–1044 (cit. on p. 14).

[Che23] Che, Tianshi, Ji Liu, Yang Zhou, Jiaxiang Ren, Jiwen Zhou, Victor Sheng, Huaiyu Dai, and Dejing Dou: 'Federated Learning of Large Language Models with Parameter-Efficient Prompt Tuning and Adaptive Optimization'. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Bouamor, Houda, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023: pp. 7871–7888 (cit. on p. 3).

[Che22] Chen, Hao, Ming Xiao, and Zhibo Pang: 'Satellite-Based Computing Networks with Federated Learning'. *IEEE Wireless Communications* (Feb. 2022), vol. 29(1): pp. 78–84 (cit. on p. 84).

[Cho20] Choe, Yo Joong, Jiyeon Ham, and Kyubyong Park: *An Empirical Study of Invariant Risk Minimization*. 2020 (cit. on p. 69).

[Coe17] Coelho, César, Sam Cooper, Mario Merri, Mehran Sarkarati, and Otto Koudelka: 'NanoSat MO Framework: Drill down your nanosatellites platform using CCSDS Mission Operations services'. Sept. 28, 2017 (cit. on p. 92).

[Coe25] Coello, Carlos A. Coello: 'Multiobjective Optimization'. *Handbook of Heuristics*. Ed. by Martí, Rafael, Panos M. Pardalos, and Mauricio G.C. Resende. Cham: Springer Nature Switzerland, 2025: pp. 231–257 (cit. on p. 3).

[Coe05] Coello, Carlos A. Coello and Nareli Cruz Cortés: 'Solving Multiobjective Optimization Problems Using an Artificial Immune System'. *Genetic Programming and Evolvable Machines* (June 2005), vol. 6(2): pp. 163–190 (cit. on pp. 29, 57).

[Cou12] Council of the European Union: *Council regulation (EU) no 1025/2012*. https://eur-lex.europa.eu/eli/reg/2012/1025/oj/eng. 2012 (cit. on p. 91).

[Cui21] Cui, Sen, Weishen Pan, Jian Liang, Changshui Zhang, and Fei Wang: 'Addressing Algorithmic Disparity and Performance Inconsistency in Federated Learning'. *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021: pp. 26091–26102 (cit. on pp. 16, 73).

[Czy98] Czyzak, Piotr and Adrezej Jaszkiewicz: 'Pareto simulated annealinga metaheuristic technique for multiple-objective combinatorial optimization'. *Journal of Multi-Criteria Decision Analysis* (1998), vol. 7(1): pp. 34–47 (cit. on p. 27).

[Dam18] Damle, Anil, Victor Minden, and Lexing Ying: 'Simple, direct and efficient multi-way spectral clustering'. *Information and Inference: A Journal of the IMA* (June 2018), vol. 8(1): pp. 181–203 (cit. on p. 44).

[Deb02] Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan: 'A fast and elitist multiobjective genetic algorithm: NSGA-II'. *IEEE Transactions on Evolutionary Computation* (2002), vol. 6(2): pp. 182–197 (cit. on p. 13).

[Den20] Denby, Bradley and Brandon Lucia: 'Orbital Edge Computing: Nanosatellite Constellations as a New Class of Computer System'. *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS 20. ACM, Mar. 2020 (cit. on p. 82).

[Den19] Denton, Remi, Ben Hutchinson, Margaret Mitchell, Timnit Gebru, and Andrew Zaldivar: 'Image Counterfactual Sensitivity Analysis for Detecting Unintended Bias'. *Proceedings of the CVPR Workshop on Fairness Accountability Transparency and Ethics in Computer Vision*. 2019 (cit. on p. 74).

[Dés12] Désidéri, Jean-Antoine: 'Multiple-gradient descent algorithm (MGDA) for multiobjective optimization'. *Comptes Rendus Mathematique* (2012), vol. 350(5): pp. 313–318 (cit. on p. 15).

[Dia20] Diao, Enmao, Jie Ding, and Vahid Tarokh: 'HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients'. *ArXiv* (2020), vol. abs/2010.01264 (cit. on p. 86).

[Dua21a] Duan, Moming, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan: 'FedGroup: Efficient Federated Learning via Decomposed Similarity-Based Clustering'. *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2021: pp. 228–237 (cit. on pp. 37, 86).

[Dua21b] Duan, Moming, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan: 'FedGroup: Efficient Federated Learning via Decomposed Similarity-Based Clustering'. *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). Sept. 2021: pp. 228–237 (cit. on p. 86).

[Dwo12] Dwork, Cynthia, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel: 'Fairness through awareness'. *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ITCS '12. Cambridge, Massachusetts: Association for Computing Machinery, 2012: pp. 214–226 (cit. on p. 72).

[ECS23] ECSS: *ECSS-E-HB-40-02A: Machine Learning Qualification for Space Applications Handbook*. Tech. rep. European Cooperation for Space Standardization (ECSS), 2023 (cit. on p. 88).

[Elm22a] Elmahallawy, Mohamed and Tie Luo: 'AsyncFLEO: Asynchronous Federated Learning for LEO Satellite Constellations with High-Altitude Platforms'. *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, Dec. 2022 (cit. on p. 85).

[Elm22b] Elmahallawy, Mohamed and Tie Luo: 'FedHAP: Fast Federated Learning for LEO Constellations using Collaborative HAPs'. *2022 14th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, Nov. 2022 (cit. on p. 85).

[Fel22] Felten, Florian and Lucas N. Alegre: *MORL-Baselines: Multi-Objective Reinforcement Learning algorithms implementations*. https://github.com/LucasAlegre/morl-baselines. 2022 (cit. on p. 27).

[Fel23] Felten, Florian, Lucas Nunes Alegre, Ann Nowe, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno Castro da Silva: 'A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning'. en. *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*. 2023 (cit. on pp. 27, 47).

[Fou21] Fourati, Fares and Mohamed-Slim Alouini: 'Artificial intelligence for satellite communication: A review'. *Intelligent and Converged Networks* (Sept. 2021), vol. 2(3): pp. 213–243 (cit. on p. 85).

[Gau23] Gauffriau, Adrien and Claire Pagetti: *Formal description of ML models for unambiguous implementation*. July 24, 2023 (cit. on p. 92).

[Gen24] Geng, Daoqu, Shouzheng Wang, and Yihang Zhang: 'MultiObjective Federated Averaging Algorithm'. *Expert Systems* (Nov. 2024), vol. (cit. on pp. 3, 14).

[Gho22a] Ghosh, Avishek, Jichan Chung, Dong Yin, and Kannan Ramchandran: 'An Efficient Framework for Clustered Federated Learning'. *IEEE Transactions on Information Theory* (Dec. 2022), vol. 68(12): pp. 8076–8091 (cit. on p. 86).

[Gho22b] Ghosh, Avishek, Jichan Chung, Dong Yin, and Kannan Ramchandran: 'An Efficient Framework for Clustered Federated Learning'. *IEEE Transactions on Information Theory* (Dec. 2022), vol. 68(12): pp. 8076–8091 (cit. on p. 86).

[Gho20] Ghosh, Avishek, Jichan Chung, Dong Yin, and Kannan Ramchandran: 'An efficient framework for clustered federated learning'. *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020 (cit. on pp. 20, 37).

[Giu22] Giuffrida, Gianluca, Luca Fanucci, Gabriele Meoni, Matej Bati, Léonie Buckley, Aubrey Dunne, Chris van Dijk, Marco Esposito, John Hefele, Nathan Vercruyssen, Gianluca Furano, Massimiliano Pastena, and Josef Aschbacher: 'The -Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation'. *IEEE Transactions on Geoscience and Remote Sensing* (2022), vol. 60: pp. 1–14 (cit. on p. 90).

[Gol11] Goldbloom, Anthony and Ben Hamner: *Heritage Health Prize*. https://kaggle.com/competitions/hhp. Kaggle. 2011 (cit. on p. 74).

[Goo20] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio: 'Generative adversarial networks'. *Communications of the ACM* (Oct. 2020), vol. 63(11): pp. 139–144 (cit. on p. 86).

[Han20] Hanzely, Filip and Peter Richtárik: 'Federated Learning of a Mixture of Global and Local Models'. *ArXiv* (2020), vol. abs/2002.05516 (cit. on p. 86).

[Har16] Hardt, Moritz, Eric Price, and Nathan Srebro: 'Equality of opportunity in supervised learning'. *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016: pp. 3323–3331 (cit. on p. 72).

[Har23a] Hartmann, Maria, Grégoire Danoy, Mohammed Alswaitti, and Pascal Bouvry: 'MOFL/D: A Federated Multi-objective Learning Framework with Decomposition'. *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*. 2023 (cit. on pp. 8, 18, 19).

[Har23b] Hartmann, Maria, Grégoire Danoy, Mohammed Alswaitti, and Pascal Bouvry: 'MOFL/D: A Federated Multi-objective Learning Framework with Decomposition'. *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*. New Orleans, LA, USA: International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023, 2023: pp. 1–13 (cit. on p. 87).

[Har24] Hartmann, Maria, Grégoire Danoy, and Pascal Bouvry: 'FedPref: Federated Learning Across Heterogeneous Multi-objective Preferences'. *ACM Trans. Model. Perform. Eval. Comput. Syst.* (Dec. 2024), vol. Just Accepted (cit. on pp. 18, 19).

[Har25a] Hartmann, Maria, Grégoire Danoy, and Pascal Bouvry: 'FedPref: Federated Learning Across Heterogeneous Multi-objective Preferences'. *ACM Trans. Model. Perform. Eval. Comput. Syst.* (May 2025), vol. 10(2) (cit. on pp. 68, 69, 73, 76).

[Har25b] Hartmann, Maria, Grégoire Danoy, and Pascal Bouvry: 'Multi-objective methods in Federated Learning: A survey and taxonomy'. *International Workshop on Federated Learning with Generative AI In Conjunction with IJCAI 2025 (FedGenAI-IJCAI'25)*. 2025 (cit. on p. 4).

[Hu22a] Hu, Zeou, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu: 'Federated Learning Meets Multi-Objective Optimization'. *IEEE Transactions on Network Science and Engineering* (July 2022), vol. 9(4). Conference Name: IEEE Transactions on Network Science and Engineering: pp. 2039–2051 (cit. on pp. 4, 69).

[Hu22b] Hu, Zeou, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu: 'Federated Learning Meets Multi-Objective Optimization'. *IEEE Transactions on Network Science and Engineering* (July 2022), vol. 9(4): pp. 2039–2051 (cit. on pp. 8, 15, 16).

[Hua23] Huang, Zhi-An, Yao Hu, Rui Liu, Xiaoming Xue, Zexuan Zhu, Linqi Song, and Kay Chen Tan: 'Federated Multi-Task Learning for Joint Diagnosis of Multiple Mental Disorders on MRI Scans'. *IEEE Transactions on Biomedical Engineering* (Apr. 2023), vol. 70(4): pp. 1137–1149 (cit. on p. 20).

[Int94] International Organization for Standardization (ISO): *ISO/IEC 7498-1:1994. Information technology Open Systems Interconnection Basic Reference Model: The Basic Model*. 1994 (cit. on p. 93).

[Izz22] Izzo, Dario, Gabriele Meoni, Pablo Gómez, Dominik Dold, and Alexander Zoechbauer: 'Selected trends in artificial intelligence for space applications'. *Artificial Intelligence for Space: AI4SPACE*. CRC Press, 2022: pp. 21–52 (cit. on pp. 82, 84).

[Jab23] Jabbarpour, Mohammad Reza, Bahman Javadi, Philip Leong, Rodrigo N. Calheiros, David Boland, and Chris Butler: 'Performance Analysis of Federated Learning in Orbital Edge Computing'. *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*. UCC 23. ACM, Dec. 2023 (cit. on p. 83).

[Jin22] Jin, Hao, Yang Peng, Wenhao Yang, Shusen Wang, and Zhihua Zhang: 'Federated Reinforcement Learning with Environment Heterogeneity'. *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. International Conference on Artificial Intelligence and Statistics. ISSN: 2640-3498. PMLR, May 3, 2022: pp. 18–37 (cit. on p. 27).

[Ju24] Ju, Li, Tianru Zhang, Salman Toor, and Andreas Hellander: 'Accelerating Fair Federated Learning: Adaptive Federated Adam'. *IEEE Transactions on Machine Learning in Communications and Networking* (2024), vol. 2: pp. 1017–1032 (cit. on pp. 15, 16, 69).

[Kai21a] Kairouz, Peter et al.: 'Advances and Open Problems in Federated Learning'. *Foundations and Trendső in Machine Learning* (2021), vol. 14(12): pp. 1–210 (cit. on pp. 10, 85, 87).

[Kai21b] Kairouz, Peter et al.: 'Advances and Open Problems in Federated Learning'. *Foundations and Trendső in Machine Learning* (2021), vol. 14(12): pp. 1–210 (cit. on p. 85).

[Kan24a] Kang, Yan, Hanlin Gu, Xingxing Tang, Yuanqin He, Yuzhu Zhang, Jinnan He, Yuxing Han, Lixin Fan, Kai Chen, and Qiang Yang: 'Optimizing Privacy, Utility, and Efficiency in a Constrained Multi-Objective Federated Learning Framework'. *ACM Trans. Intell. Syst. Technol.* (Dec. 2024), vol. 15(6) (cit. on pp. 15, 16).

[Kan24b] Kang, Yan, Ziyao Ren, Lixin Fan, Linghua Yang, Yongxin Tong, and Qiang Yang: *Hyperparameter Optimization for SecureBoost via Constrained Multi-Objective Federated Learning*. Apr. 6, 2024 (cit. on pp. 13, 15).

[Kar20] Karimireddy, Sai Praneeth, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh: 'SCAFFOLD: Stochastic Controlled Averaging for Federated Learning'. *Proceedings of the 37th International Conference on Machine Learning*. Ed. by III, Hal Daumé and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020: pp. 5132–5143 (cit. on p. 37).

[Kar19] Karimireddy, Sai Praneeth, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh: 'SCAFFOLD: Stochastic Controlled Averaging for Federated Learning'. *International Conference on Machine Learning*. 2019 (cit. on p. 3).

[Kim22] Kim, Jinkyu, Geeho Kim, and Bohyung Han: 'Multi-Level Branched Regularization for Federated Learning'. *ArXiv* (2022), vol. abs/2207.06936 (cit. on p. 86).

[Kin24] Kinoshita, Takato, Naoki Masuyama, and Yusuke Nojima: 'A Federated Data-driven Multiobjective Evolutionary Algorithm via Continual Learnable Clustering'. *2024 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, June 2024: pp. 1–7 (cit. on pp. 18, 69).

[Kus17] Kusner, Matt, Joshua Loftus, Chris Russell, and Ricardo Silva: 'Counterfactual fairness'. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017: pp. 4069–4079 (cit. on p. 72).

[Le 22] Le Quy, Tai, Arjun Roy, Vasileios Iosifidis, Wenbin Zhang, and Eirini Ntoutsi: 'A survey on datasets for fairness-aware machine learning'. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2022), vol. 12(3): e1452 (cit. on p. 74).

[Li21a] Li, Anran, Lan Zhang, Juntao Tan, Yaxuan Qin, Junhao Wang, and Xiang-Yang Li: 'Sample-level Data Selection for Federated Learning'. *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*. IEEE, May 2021 (cit. on p. 85).

[Li21b] Li, Li, Moming Duan, Duo Liu, Yu Zhang, Ao Ren, Xianzhang Chen, Yujuan Tan, and Chengliang Wang: 'FedSAE: A Novel Self-Adaptive Federated Learning Framework in Heterogeneous Systems'. *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2021 (cit. on p. 86).

[Li21c] Li, Q., B. He, and D. Song: 'Model-Contrastive Federated Learning'. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2021: pp. 10708–10717 (cit. on p. 37).

[Li18] Li, Tian, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith: *Federated Optimization in Heterogeneous Networks*. 2018 (cit. on pp. 37, 48, 127, 128).

[Li20a] Li, Tian, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith: 'Federated optimization in heterogeneous networks'. *Proceedings of Machine learning and systems* (2020), vol. 2: pp. 429–450 (cit. on p. 76).

[Li20b] Li, Tian, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith: 'Fair Resource Allocation in Federated Learning'. *International Conference on Learning Representations*. 2020 (cit. on p. 16).

[Lin19] Lin, Xi, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong: 'Pareto multi-task learning'. *Advances in neural information processing systems* (2019), vol. 32 (cit. on p. 69).

[Liu21] Liu, S. and L. N. Vicente: 'The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning'. *Annals of Operations Research* (Mar. 17, 2021), vol. (cit. on p. 22).

[Liu24] Liu, Yang, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang: 'Vertical Federated Learning: Concepts, Advances, and Challenges'. *IEEE Transactions on Knowledge and Data Engineering* (2024), vol.: pp. 1–20 (cit. on p. 86).

[Liu15] Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang: 'Deep Learning Face Attributes in the Wild'. *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015 (cit. on pp. 69, 74).

[Loh20] Lohaus, Michael, Michael Perrot, and Ulrike Von Luxburg: 'Too Relaxed to Be Fair'. *Proceedings of the 37th International Conference on Machine Learning*. Ed. by III, Hal Daumé and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020: pp. 6360–6369 (cit. on p. 73).

[Lon22] Long, Guodong, Ming Xie, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang: 'Multi-center federated learning: clients clustering for better personalization'. *World Wide Web* (June 2022), vol. 26(1): pp. 481–500 (cit. on p. 37).

[Ma22] Ma, Xiaosong, Jie Zhang, Song Guo, and Wenchao Xu: 'Layer-wised Model Aggregation for Personalized Federated Learning'. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022 (cit. on p. 86).

[McM17a] McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas: 'Communication-Efficient Learning of Deep Networks from Decentralized Data'. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*. Ed. by Singh, Aarti and Xiaojin (Jerry) Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 2017: pp. 1273–1282 (cit. on p. 83).

[McM17b] McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas: 'Communication-Efficient Learning of Deep Networks from Decentralized Data'. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Singh, Aarti and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, Apr. 2017: pp. 1273–1282 (cit. on pp. 9, 23).

[McM17c] McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas: 'Communication-Efficient Learning of Deep Networks from Decentralized Data'. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Singh, Aarti and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 2017: pp. 1273–1282 (cit. on p. 27).

[McM17d] McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas: 'Communication-Efficient Learning of Deep Networks from Decentralized Data'. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Singh, Aarti and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, Apr. 2017: pp. 1273–1282 (cit. on p. 73).

[Meh22] Mehrabi, Ninareh, Cyprien de Lichy, John McKay, Cynthia He, and William Campbell: *Towards Multi-Objective Statistically Fair Federated Learning*. Jan. 24, 2022 (cit. on pp. 4, 8, 15, 16, 26, 69, 73).

[Meh21] Mehrabi, Ninareh, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan: 'A Survey on Bias and Fairness in Machine Learning'. *ACM Comput. Surv.* (July 13, 2021), vol. 54(6): 115:1–115:35 (cit. on pp. 71, 72).

[Mil20] Milojkovic, Nikola, Diego Antognini, Giancarlo Bergamin, Boi Faltings, and Claudiu Musat: 'Multi-Gradient Descent for Multi-Objective Recommender Systems'. *CoRR* (2020), vol. abs/2001.00846 (cit. on p. 137).

[Mni15] Mnih, Volodymyr et al.: 'Human-level control through deep reinforcement learning'. *Nature* (Feb. 2015), vol. 518(7540): pp. 529–533 (cit. on p. 47).

[Moh19] Mohri, Mehryar, Gary Sivek, and Ananda Theertha Suresh: 'Agnostic Federated Learning'. *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Chaudhuri, Kamalika and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019: pp. 4615–4625 (cit. on p. 16).

[Mor23] Morales-Hernández, Alejandro, Inneke Van Nieuwenhuyse, and Sebastian Rojas Gonzalez: 'A survey on multi-objective hyperparameter optimization algorithms for machine learning'. *Artificial Intelligence Review* (Aug. 1, 2023), vol. 56(8): pp. 8043–8093 (cit. on p. 22).

[Mor24] Morell, José Ángel, Zakaria Abdelmoiz Dahi, Francisco Chicano, Gabriel Luque, and Enrique Alba: 'A multi-objective approach for communication reduction in federated learning under devices heterogeneity constraints'. *Future Generation Computer Systems* (June 2024), vol. 155: pp. 367–383 (cit. on p. 14).

[Muñ19] Muñoz-González, Luis, Kenneth T. Co, and Emil C. Lupu: 'Byzantine-Robust Federated Machine Learning through Adaptive Model Averaging'. *ArXiv* (2019), vol. abs/1909.05125 (cit. on p. 87).

[Nga05] Ngatchou, P., Anahita Zarei, and A. El-Sharkawi: 'Pareto Multi Objective Optimization'. *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*. Vol. 2005. Dec. 2005: pp. 84–91 (cit. on p. 28).

[Ngu22] Nguyen, John, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat: 'Where to Begin? On the Impact of Pre-Training and Initialization in Federated Learning'. Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022). Nov. 22, 2022 (cit. on p. 25).

[Nis19] Nishio, Takayuki and Ryo Yonetani: 'Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge'. *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. IEEE, May 2019 (cit. on p. 86).

[Gro25] Group, Khronos. https://registry.khronos.org/NNEF. Last accessed 2025-09-21. 2025 (cit. on p. 91).

[com25] community, ONNX. https://onnx.ai/onnx/index.html. Last accessed 2025-09-21. 2025 (cit. on p. 91).

[Int25] Intel. https://docs.openvino.ai/2025/index.html. Last accessed 2025-09-22. 2025 (cit. on p. 91).

[Öst23] Östman, Johan, Pablo Gomez, Vinutha Magal Shreenath, and Gabriele Meoni: *Decentralised Semi-supervised Onboard Learning for Scene Classification in Low-Earth Orbit*. 2023 (cit. on p. 85).

[Pad21] Padh, Kirtan, Diego Antognini, Emma Lejal-Glaude, Boi Faltings, and Claudiu Musat: 'Addressing fairness in classification with a model-agnostic multi-objective algorithm'. *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*. Ed. by Campos, Cassio de and Marloes H. Maathuis. Vol. 161. Proceedings of Machine Learning Research. PMLR, July 2021: pp. 600–609 (cit. on pp. 73, 137).

[Pes22] Pessach, Dana and Erez Shmueli: 'A Review on Fairness in Machine Learning'. *ACM Comput. Surv.* (Feb. 2022), vol. 55(3) (cit. on pp. 72, 74).

[Qi21] Qi, Jiaju, Qihao Zhou, Lei Lei, and Kan Zheng: *Federated Reinforcement Learning: Techniques, Applications, and Open Challenges*. Oct. 24, 2021 (cit. on p. 26).

[Raf20] Raffin, Antonin: *RL Baselines3 Zoo*. https://github.com/DLR-RM/rl-baselines3-zoo. 2020 (cit. on p. 28).

[Raf21] Raffin, Antonin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann: 'Stable-Baselines3: Reliable Reinforcement Learning Implementations'. *Journal of Machine Learning Research* (2021), vol. 22(268): pp. 1–8 (cit. on p. 27).

[Raz22] Razmi, Nasrin, Bho Matthiesen, Armin Dekorsy, and Petar Popovski: 'On-Board Federated Learning for Dense LEO Constellations'. *ICC 2022 - IEEE International Conference on Communications*. IEEE, May 2022 (cit. on p. 84).

[Raz24] Razmi, Nasrin, Bho Matthiesen, Armin Dekorsy, and Petar Popovski: 'On-board Federated Learning for Satellite Clusters with Inter-Satellite Links'. *IEEE Transactions on Communications* (2024), vol.: pp. 1–1 (cit. on pp. 84, 85, 87).

[Rei23a] Reiff, Jean-Marie, Jean-Philippe Humbert, Pascal Bouvry, Grégoire Danoy, Mohammed Alswaitti, Manuel Combarro Simón, Maria Hartmann, Hedieh Haddad, Lucas Cicero, Jean Lancrenon, Nicolas Domenjoud, Leslie Fouqueray, Natalia Vinogradova, and Ruddy Enguehard: *Trustworthiness in ICT, Aerospace, and Construction applications*. Tech. rep. Luxembourg Institute of Standardisation, Accreditation, Safety, Quality of Products, and Services (ILNAS), 2023 (cit. on p. 88).

[Rei23b] Reiff, Jean-Marie, Jean-Philippe Humbert, Pascal Bouvry, Grégoire Danoy, Mohammed Alswaitti, Manuel Combarro Simón, Maria Hartmann, Hedieh Haddad, Lucas Cicero, Jean Lancrenon, Nicolas Domenjoud, Leslie Fouqueray, Natalia Vinogradova, and Ruddy Enguehard: *Trustworthiness in ICT, Aerospace and Construction Applications*. Institut luxembourgeois de la normalisation, de l'accréditation, de la sécurité et qualité des produits et services (ILNAS), 2023 (cit. on p. 89).

[Riq15] Riquelme, Nery, Christian Von Lucken, and Benjamin Baran: 'Performance metrics in multi-objective optimization'. *2015 Latin American Computing Conference (CLEI)*. Arequipa, Peru: IEEE, Oct. 2015: pp. 1–11 (cit. on p. 28).

[Rod23] Rodríguez-Barroso, Nuria, Daniel Jiménez-López, M. Victoria Luzón, Francisco Herrera, and Eugenio Martínez-Cámara: 'Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges'. *Information Fusion* (Feb. 2023), vol. 90: pp. 148–173 (cit. on p. 87).

[Sat19] Sattler, Felix, Klaus-Robert Müller, and Wojciech Samek: 'Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints'. *IEEE Transactions on Neural Networks and Learning Systems* (2019), vol. 32: pp. 3710–3722 (cit. on pp. 25, 37, 40, 42, 48, 49, 76, 127, 128, 135).

[Sen24] Sen, Pritam and Cristian Borcea: 'FedMTL: Privacy-Preserving Federated Multi-Task Learning'. *ECAI 2024*. IOS Press, Oct. 2024 (cit. on pp. 18, 20).

[Sen18] Sener, Ozan and Vladlen Koltun: 'Multi-task learning as multi-objective optimization'. *Advances in neural information processing systems* (2018), vol. 31 (cit. on pp. 68, 71).

[Sha22] Sharma, Shubhkirti and Vijay Kumar: 'A Comprehensive Review on Multi-objective Optimization Techniques: Past, Present and Future'. *Archives of Computational Methods in Engineering* (Nov. 1, 2022), vol. 29(7): pp. 5605–5633 (cit. on pp. 23, 39).

[She25] Shen, Yuhao, Wei Xi, Yunyun Cai, Yuwei Fan, He Yang, and Jizhong Zhao: 'Multi-objective federated learning: Balancing global performance and individual fairness'. *Future Generation Computer Systems* (Jan. 2025), vol. 162: p. 107468 (cit. on p. 16).

[Shi24] Shi, Yuxin, Han Yu, and Cyril Leung: 'Towards Fairness-Aware Federated Learning'. *IEEE Transactions on Neural Networks and Learning Systems* (2024), vol.: pp. 1–17 (cit. on p. 87).

[Smi18] Smith, Virginia, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar: *Federated Multi-Task Learning*. Feb. 27, 2018 (cit. on p. 87).

[So22] So, Jinhyun, Kevin Hsieh, Behnaz Arzani, Shadi Noghabi, Salman Avestimehr, and Ranveer Chandra: *FedSpace: An Efficient Federated Learning Framework at Satellites and Ground Stations*. 2022 (cit. on pp. 84, 85).

[Spa23] Space Data Systems (CCSDS), The Consultative Committee for: *CCSDS 130.0-G-4: Overview of Space Communications Protocols*. Tech. rep. The Consultative Committee for Space Data Systems (CCSDS), 2023 (cit. on p. 93).

[Spa21] Space Data Systems (CCSDS), The Consultative Committee for: *CCSDS 401.0-B-32: Radio Frequency Modulation Systems - Part 1: Earth Stations and Spacecraft*. Tech. rep. The Consultative Committee for Space Data Systems (CCSDS), 2021 (cit. on p. 93).

[Súk22] Súkeník, Peter and Christoph H. Lampert: *Generalization In Multi-Objective Machine Learning*. 2022 (cit. on p. 22).

[Sun19] Sun, Ziteng, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan: *Can You Really Backdoor Federated Learning?* 2019 (cit. on p. 87).

[Tal09] Talbi, El-Ghazali: *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009 (cit. on p. 11).

[Tan23] Tan, Alysa Ziying, Han Yu, Lizhen Cui, and Qiang Yang: 'Towards Personalized Federated Learning'. *IEEE Transactions on Neural Networks and Learning Systems* (Dec. 2023), vol. 34(12): pp. 9587–9603 (cit. on pp. 19, 37, 86).

[Vam11] Vamplew, Peter, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker: 'Empirical evaluation methods for multiobjective reinforcement learning algorithms'. *Machine Learning* (July 1, 2011), vol. 84(1): pp. 51–80 (cit. on pp. 27, 47).

[Var22] Varrette, S., H. Cartiaux, S. Peter, E. Kieffer, T. Valette, and A. Olloh: 'Management of an Academic HPC & Research Computing Facility: The ULHPC Experience 2.0'. *Proc. of the 6th ACM High Performance Computing and Cluster Technologies Conf. (HPCCT 2022)*. Fuzhou, China: Association for Computing Machinery (ACM), 2022 (cit. on p. 124).

[Wan19] Wang, Kangkang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage: *Federated Evaluation of On-device Personalization*. 2019 (cit. on p. 48).

[Wig98] Wightman, Linda F: 'LSAC National Longitudinal Bar Passage Study.' *LSAC Research Report Series* (1998), vol. (cit. on p. 74).

[Wu22] Wu, Chenrui, Yifei Zhu, and Fangxin Wang: 'DSFL: Decentralized Satellite Federated Learning for Energy-Aware LEO Constellation Computing'. *2022 IEEE International Conference on Satellite Computing (Satellite)*. IEEE, Nov. 2022 (cit. on p. 85).

[Wu23] Wu, Lingling and Jingjing Zhang: 'FedGSM: Efficient Federated Learning for LEO Constellations with Gradient Staleness Mitigation'. *2023 IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, Sept. 2023 (cit. on pp. 84, 85).

[Xu20] Xu, Jie, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik: 'Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control'. *Proceedings of the 37th International Conference on Machine Learning*. Ed. by III, Hal Daumé and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020: pp. 10607–10616 (cit. on pp. 29, 57).

[Xu23] Xu, Xiaolong, Haoyuan Li, Zheng Li, and Xiaokang Zhou: 'Safe: Synergic Data Filtering for Federated Learning in Cloud-Edge Computing'. *IEEE Transactions on Industrial Informatics* (Feb. 2023), vol. 19(2): pp. 1655–1665 (cit. on p. 85).

[Yan23a] Yang, Fangjie, Honglan Huang, Wei Shi, Yang Ma, Yanghe Feng, Guangquan Cheng, and Zhong Liu: 'PMDRL: Pareto-front-based multi-objective deep reinforcement learning'. *Journal of Ambient Intelligence and Humanized Computing* (Sept. 1, 2023), vol. 14(9): pp. 12663–12672 (cit. on p. 22).

[Yan23b] Yang, Haibo, Zhuqing Liu, Jia Liu, Chaosheng Dong, and Michinari Momma: 'Federated Multi-Objective Learning'. *Thirty-seventh Conference on Neural Information Processing Systems*. 2023 (cit. on pp. 8, 18, 23, 68, 73, 87).

[Yan19] Yang, Qiang, Yang Liu, Tianjian Chen, and Yongxin Tong: 'Federated Machine Learning: Concept and Applications'. *ACM Trans. Intell. Syst. Technol.* (2019), vol. 10(2) (cit. on pp. 23, 86).

[Ye23a] Ye, Mang, Xiuwen Fang, Bo Du, Pong C. Yuen, and Dacheng Tao: 'Heterogeneous Federated Learning: State-of-the-art and Research Challenges'. *ACM Comput. Surv.* (Oct. 2023), vol. 56(3) (cit. on p. 37).

[Ye23b] Ye, Mang, Xiuwen Fang, Bo Du, Pong C. Yuen, and Dacheng Tao: 'Heterogeneous Federated Learning: State-of-the-art and Research Challenges'. *ACM Computing Surveys* (Oct. 2023), vol. 56(3): pp. 1–44 (cit. on pp. 85, 86).

[Ye22] Ye, Mao and Qiang Liu: 'Pareto navigation gradient descent: a first-order algorithm for optimization in pareto set'. *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Ed. by Cussens, James and Kun Zhang. Vol. 180. Proceedings of Machine Learning Research. PMLR, Aug. 2022: pp. 2246–2255 (cit. on p. 16).

[Yeh09] Yeh, I-Cheng and Che-Hui Lien: 'The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients'. *Expert Syst. Appl.* (2009), vol. 36: pp. 2473–2480 (cit. on p. 74).

[Yin23] Yin, Benshun, Zhiyong Chen, and Meixia Tao: 'Predictive GAN-Powered Multi-Objective Optimization for Hybrid Federated Split Learning'. *IEEE Transactions on Communications* (Aug. 2023), vol. 71(8): pp. 4544–4560 (cit. on p. 14).

[Yin18] Yin, Dong, Yudong Chen, Ramchandran Kannan, and Peter Bartlett: 'Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates'. *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Dy, Jennifer and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018: pp. 5650–5659 (cit. on p. 87).

[Yoo21] Yoon, Tehrim, Sumin Shin, Sung Ju Hwang, and Eunho Yang: 'FedMix: Approximation of Mixup under Mean Augmented Federated Learning'. *ArXiv* (2021), vol. abs/2107.00233 (cit. on p. 86).

[Zha23] Zhao, Fangyuan, Xuebin Ren, Shusen Yang, Peng Zhao, Rui Zhang, and Xinxin Xu: 'Federated multi-objective reinforcement learning'. *Information Sciences* (2023), vol. 624: pp. 811–832 (cit. on p. 26).

[Zhu20a] Zhu, Hangyu and Yaochu Jin: 'Multi-Objective Evolutionary Federated Learning'. *IEEE Transactions on Neural Networks and Learning Systems* (Apr. 2020), vol. 31(4): pp. 1310–1322 (cit. on pp. 8, 14).

[Zhu22] Zhu, Hangyu and Yaochu Jin: 'Real-Time Federated Evolutionary Neural Architecture Search'. *IEEE Transactions on Evolutionary Computation* (2022), vol. 26(2): pp. 364–378 (cit. on pp. 4, 15, 17).

[Zhu21a] Zhu, Hangyu, Haoyu Zhang, and Yaochu Jin: 'From federated learning to federated neural architecture search: a survey'. *Complex & Intelligent Systems* (Jan. 2021), vol. 7(2): pp. 639–657 (cit. on pp. 3, 14).

[Zhu21b] Zhu, Zhuangdi, Junyuan Hong, and Jiayu Zhou: 'Data-Free Knowledge Distillation for Heterogeneous Federated Learning'. *Proceedings of machine learning research* (July 2021), vol. 139: pp. 12878–12889 (cit. on p. 86).

[Zhu20b] Zhuo, Hankz Hankui, Wenfeng Feng, Yufeng Lin, Qian Xu, and Qiang Yang: *Federated Deep Reinforcement Learning*. Feb. 9, 2020 (cit. on p. 26).

[Zit99] Zitzler, E. and L. Thiele: 'Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach'. *IEEE Transactions on Evolutionary Computation* (1999), vol. 3(4): pp. 257–271 (cit. on p. 57).

[Zit03] Zitzler, Eckart, Lothar Thiele, Marco Laumanns, C.M. Fonseca, and Viviane Fonseca: 'Performance Assessment of Multiobjective Optimizers: An Analysis and Review'. *Evolutionary Computation, IEEE Transactions on* (May 2003), vol. 7: pp. 117–132 (cit. on p. 28).

A | Parameters and complementary results

A.1 | MOFL/D: A framework for federated multi-objective learning

A.1.1 | Complete experimental parameters

All experiments with two or three clients were repeated 10 times each, with respective seeds $5, 11, 17, 176, 462, 488, 3011, 6543, 9347, 675234$. Experiments with five clients were repeated only 5 times due to the high computing cost; these experiments were run with seeds $5, 17, 488, 3011, 6543$. The number of runs on the non-federated system is matched to the total number of clients involved in all repetitions of the federated system, so e.g. $2 \cdot 10 = 20$ to compare with a federated system with two clients repeated 10 times. Note that our implementation uses multi-threading to model individual federated participants; therefore the experiments are not deterministic and will not reproduce precisely the same numerical results.

A.1.2 | Computing details

The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [Var22] – see https://hpc.uni.lu. The computing time equates to approximately 1450 hours (i.e., more than 60 days) for a single HPC node. The technical specifications of a cluster compute node are given in Table A.2

A.1.3 | Additional Results

A.1.3.1 | Using pre-trained models

The experiments do not offer conclusive results for or against the use of pre-trained models, obtained earlier in the optimisation process, to initialise new federated learning runs.
In some cases, e.g. in the results for the Deep-Sea Treasure environment and the Lunar Lander environment shown in Fig. A.1(b) and Fig. A.2(a), respectively, the results of the algorithm run with pre-trained models seem to match or at times during the optimisation process even outperform the algorithm run without pre-trained models. Also notable in some cases, e.g. in the results shown for the DST environment, is the significantly reduced variance of the hypervolume obtained by the system with pre-trained models in the initial stages of convergence, as well as the slightly faster increase of the hypervolume. However, when comparing the corresponding values of the sparsity metric in Table 3.2, it becomes apparent that these are significantly higher when pre-trained models are used. This indicates that this instantiation of the algorithm tends to find more solutions that are in close proximity to ones already discovered, leading to a high number of solutions, but

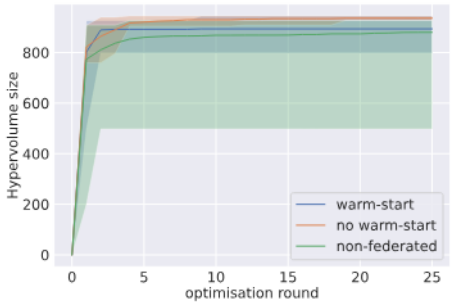
Table A.1: The full set of hyperparameters for all experiments presented in this paper. Left to right: Deep-Sea Treasure (DST), Multi-objective Lunar Lander (MO-LL) and Deterministic Minecart (DMC).

Parameter name	**DST**	**MO-LL**	**DMC**
Metaheuristic (Pareto Simulated Annealing)			
Iterations	20	25	25
Samples per round	5	10	10
Federated Learning (FedAvg)			
Total iterations	10^5	10^5	$1.5 \cdot 10^5$
Iterations/local round	$(2/5/10) \cdot 10^3$	$(2/5/10) \cdot 10^3$	$(2/5/10) \cdot 10^3$
Number of clients	2/3/5	2/3/5	2/3/5
Reinforcement Learning (DQN)			
Train frequency	16	4	32
Gradient steps	8	-1	32
Gamma	0.98	0.99	0.99
Exploration fraction	0.2	0.12	0.8
Exploration final episode	$7 \cdot 10^{-2}$	0.1	$5 \cdot 10^{-2}$
Target update interval	600	250	750
Buffer size	10^4	$5 \cdot 10^4$	$5 \cdot 10^4$
Batch size	128	64	64
Learning rate	$4 \cdot 10^{-3}$	$6.3 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
Network	[256,256]	[256,256]	[256,256]
Reference point	$(0, -50)$	$(-200, -200, -200, -200)$	$(-1, -1, -200)$

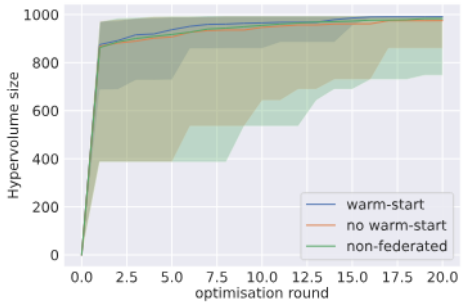
Table A.2: Hardware specifications of the cluster nodes employed for experiments.

CPU	2 AMD Epyc ROME 7H12 @ 2.6 GHz [64c/280W]
RAM	256GB

with low diversity. This observation also serves to explain the reduced performance on the Deterministic Minecart environment, as optimal solutions in this environment are sparse. Therefore, any attempt to exploit the neighbourhood of a previous solution is less likely to be successful.

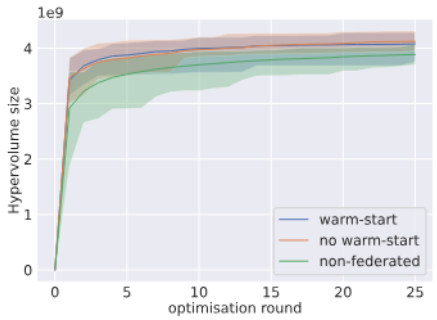


(a) Results for the Deterministic Minecart environment.

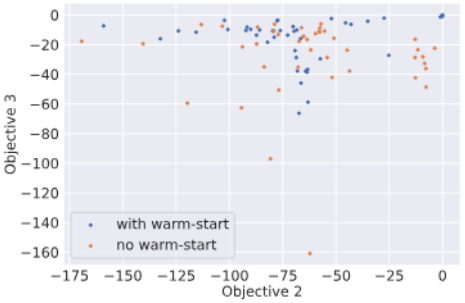


(b) Results for the Deep-Sea Treasure environment.

Figure A.1: Hypervolume evolution compared for experiments run with and without pre-trained models. The duration of the local training phase in federation was fixed at 5000 iterations; the number of federated clients was fixed at 3.



(a) The evolution of hypervolumes obtained using pre-trained models and those obtained without pre-trained models are quite similar.



(b) Solutions obtained for objectives 2 and 3 (solution vectors projected into the plane). Solutions obtained using pre-trained models are clustered closer together than those obtained without pre-trained models.

Figure A.2: Results for experiments run on the Lunar Lander environment with and without pre-trained models. The duration of the local training phase in federation was fixed at 10000 iterations; the number of federated clients was 3.

A.2 | FedPref: solving federated multi-objective learning under preference heterogeneity

A.2.1 | Details of experiment configurations

A.2.1.1 | Hyperparameter tuning

We perform an initial hyperparameter search for all algorithms and environments, with all evaluated parameter values listed in Table A.5. Each configuration was run five times and five different randomly-generated preference distributions. For all environments except the MO-Halfcheetah environment, all experiments were performed with 20 simulated clients per system. Due to the comparably high computational cost of training on the MO-Halfcheetah

environment, the number of clients was reduced to 10 clients per system in this case. The five preference distributions remained fixed across all hyperparameter configurations to promote the comparability of results. The metric used to assess performance was the mean linearised reward obtained by the clients using their personalised preference weights. The parameter values selected as a result of the hyperparameter tuning are given in Table A.3 and A.4.

Table A.3: Complete list of parameter configurations tested during hyperparameter tuning of DQN algorithms.

		MO-LL	DMC	DST	Comment
No comm.	-	-	-	-	No federated parameters.
FedAvg	Local iterations	$(2, 5, 10 \cdot 10^3$	$(2, 5, 10) \cdot 10^3$	$(5, 10, 15) \cdot 10^2$	
FedProx	Local iterations	$(2, 5, 10 \cdot 10^3$	$(2, 5, 10) \cdot 10^3$	$(5, 10, 15) \cdot 10^2$	
	Proximal term μ	$0.01, 0.1, 1$	$0.01, 0.1, 1$	$0.01, 0.1, 1$	Based on [Li18]
CFL	Local iterations	$(2, 5, 10 \cdot 10^3$	$(2, 5, 10) \cdot 10^3$	$(5, 10, 15) \cdot 10^2$	
	Clustering threshold	$2.5, 5, 7.5$	$2, 3, 5$	$2.5, 5, 7.5$	See [a]
	Patience	$1, 2$	$1, 2$	$1, 2$	See [b]
MaTFL	Local iterations	$(2, 5, 10 \cdot 10^3$	$(2, 5, 10) \cdot 10^3$	$(5, 10, 15) \cdot 10^2$	
	Voting clients k	$5, 8, 10$	$5, 8, 10$	$5, 8, 10$	Following [Cai23]
Ours	Local iterations	$(2, 5, 10 \cdot 10^3$	$(2, 5, 10) \cdot 10^3$	$(5, 10, 15) \cdot 10^2$	
	Clustering threshold	$2.5, 5, 7.5$	$2, 3, 5$	$2.5, 5, 7.5$	Same as for CFL
	Patience	$1, 2$	$1, 2$	$1, 2$	Same as for CFL
	Min. similarity	$-1, 0$	$-1, 0$	$-1, 0$	Used during aggregation[c]

[a] Based on max. observed gradient magnitude ,as suggested in [Sat19].
[b] Rounds below threshold before clustering triggered. Introduced by us to handle slow initial gradient ramp-up.
[c] See Section 4.2.3 in the main paper for explanation.

A.2.1.2 | MORL environment parameters

Where available, the hyperparameters for the three MORL environments used in our experiments were obtained from published benchmark configurations. Where no such configurations were available, they were obtained by manual tuning. All modified parameters are reported below, in Tables A.6, and A.7. Parameters that are not listed can be assumed to be set to the default setting, as implemented in the DQN and DDPG algorithms, respectively, of the stable-baselines3 package.

A.2.1.3 | Computing resources

The number of experiments presented in this paper amounts to 4125 individual experimental runs. This corresponds to a total runtime of approximately 8560 hours on a single node of the computing cluster available to us.

Table A.4: Complete list of parameter configurations tested during hyperparameter tuning of DDPG algorithms.

		MO-LLcont.	MO-HC	Comment
No comm.	-	-	-	No federated parameters.
FedAvg	Local iterations	$5000, 10000, 15000$	$25000, 37500, 50000$	
FedProx	Local iterations	$5000, 10000, 15000$	$25000, 37500, 50000$	
	Proximal term μ	$0.01, 0.1, 1$	$0.01, 0.1, 1$	Following [Li18]
CFL	Local iterations	$5000, 10000, 15000$	$25000, 37500, 50000$	
	Clustering threshold	$10, 15, 20$	$20, 30, 40$	Based on max. observed gradient magnitude[a]
	Patience	$1, 2$	$1, 2$	Rounds below threshold before clustering triggered[b]
MaTFL	Local iterations	$5000, 10000, 15000$	$25000, 37500, 50000$	
	Voting clients k	$5, 8, 10$	$3, 5, 8$	Following [Cai23]
Ours	Local iterations	$5000, 10000, 15000$	$25000, 37500, 50000$	
	Clustering threshold	$10, 15, 20$	$20, 30, 40$	Same as for CFL
	Patience	$1, 2$	$1, 2$	Same as for CFL
	Min. similarity	$-1, 0$	$-1, 0$	Used in computing aggregation weights[c]

[a] As suggested in [Sat19].
[b] Introduced by us to handle slow initial gradient ramp-up.
[c] See Section 4.2.3 in the main paper for explanation.

Table A.5: Parameter configurations selected for each algorithm following hyperparameter tuning.

		MO-LL	DMC	DST	MO-HC	MO-LLc.
No comm.	-	-	-	-	-	-
FedAvg	Number local iterations	5000	5000	500	25000	5000
FedProx	Number local iterations	5000	5000	500	25000	5000
	Proximal term μ	1	1	1	0.01	0.01
CFL	Number local iterations	10000	2000	1000	25000	5000
	Clustering threshold	5	2	5	30	15
	Patience	2	2	2	1	2
MaTFL	Number local iterations	2000	5000	1000	25000	5000
	Number voting clients k	10	10	10	8	10
FedPref (ours)	Number local iterations	5000	5000	500	25000	10000
	Clustering threshold	5	3	5	30	15
	Patience	1	2	2	2	2
	Minimum similarity	-1	0	-1	-1	-1

Table A.6: Set of parameters used for the local training of the MO-Lunar Lander, Deterministic Minecart and Deep-Sea Treasure environments, using the DQN algorithm.

Parameter name	MO-Lunar Lander	Det. Minecart	Deep-Sea Treasure
env	mo-lunar-lander-v2	minecart-deterministic-v0	deep-sea-treasure-v0
policy	MlpPolicy	MlpPolicy	MlpPolicy
learning_rate	0.00063	0.0002	0.004
batch_size	64	64	128
buffer_size	50000	50000	10000
learning_starts	0	50000	1000
gamma	0.99	0.99	0.98
target_update_interval	250	750	600
train_freq	4	32	16
gradient_steps	-1	32	8
exploration_fraction	0.12	0.8	0.2
exploration_final_eps	0.1	0.05	0.07
net_arch	$[256, 256]$	[256, 256]	[256, 256]

Table A.7: Set of parameters used for the local training of the MO-Halfcheetah and MO-Lunar Lander continuous environment.

Parameter name	MO-Halfcheetah	MO-Lunar Lander cont.
env	mo-halfcheetah-v4	mo-lunar-lander-continuous-v2
noise_std	0.1	0.1
policy	MlpPolicy	MlpPolicy
learning_rate	0.001	0.001
buffer_size	200000	200000
learning_starts	-	10000
gamma	0.98	0.98
train_freq	1	1
gradient_steps	1	-
net_arch	$[400, 300]$	$[400, 300]$

A.2.2 | Supplementary experimental results

In this section, we include supplementary numerical results and plots that exceeded the scope of the main body of the thesis.

A.2.2.1 | Impact of topR parameter and similarity bound

This section lists numerical experimental results for the parameter sensitivity analyses carried out in Chapter 4; these same results are presented there in visual form, with some numbers quoted. We refer the reader to the relevant section in the main paper for the analysis and discussion of these results. Table A.9 contains the results for the sensitivity analysis of the *topR* parameter; Table A.8 shows the results for the analysis of the minimum similarity threshold in aggregation. All experiments were carried out with 10 different random seeds, on preference weights drawn from a Dirichlet distribution. Experiments on

all environments were run on systems of 20 federated clients, with the exception of the MO-Halfcheetah environment, which was restricted to systems of 10 clients due to its high computational cost.

A.2.2.2 | FedPref clustering validation

In this section, we show and briefly discuss additional results of the clustering validation experiments.

MO-Lunar Lander. Fig. A.3 shows the similarity of clients at three training stages during training in the MO-LL environment on an unbalanced preference distribution. Three groups with distinct similarity are clearly recognisable from the earliest stages of the training process; these correspond to the sets of clients that have been assigned the same preferences, with two such sets evidently grouped together. Later stages show the gradual separation of the different sets, likely through the clustering process. However, the two client sets that showed a high similarity from the beginning (both contained in the largest, top-left block in the figure) appear to remain in the same cluster until the end of the training process, never being separated. This could indicate either that the two different preference weights assigned to the two sets are naturally compatible during the training process, or that the FedPref algorithm might sometimes struggle to fully separate incompatible sets of clients before they converge to a local optimum. The latter could also be a consequence of the imbalanced distribution of potentially incompatible clients in this case; perhaps a small number of incompatible clients is 'dominated' by the remaining large number of compatible clients in the same cluster.

Det. Minecart. Fig. A.4 and Fig. A.5 show client similarities during training on the Det. Minecart environment with the balanced and unbalanced distribution of preferences, respectively. These results also illustrate the challenges of this environment that were discussed in the main part of the paper: the sparse reward space appears to make it difficult to reliably discover client similarities during the clustering process. We observe in both figures that clients never reach high levels of similarity as seen in the results of the MO-LL environment; it is likely that this also impedes the clustering process, leading to a suboptimal grouping into clusters. However, some successful collaboration appears to take place, as evidences by the darker-coloured patches in the middle and right images in both figures. This matches our experimental conclusions in the main paper, that the

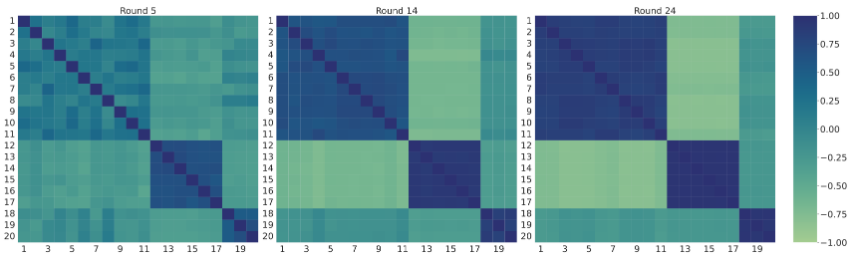


Figure A.3: Mutual client similarity at different stages during a single experimental run on the MO-LL environment, with unbalanced preference distribution. Left to right: client similarities after aggregation round 5, 14 and 24 of 28, respectively.

Table A.8: Numerical results for minimum-similarity sensitivity analysis visualised in the main part of the paper. All configurations except those on the MO-HC environment were run with 10 different random seeds across 20 clients per run. Due to the higher computational cost of solving the MO-HC environment, experiments on this environment were restricted to 10 clients per run, also for 10 runs per configuration.

Threshold	MO-LL	DMC	DST	MO-HC	MO-LLc.
−1.0	**32.22** σ11.3	−1.91σ1.0	4.41σ1.7	2980.26σ784.4	32.17σ7.0
−0.8	31.74σ11.3	−2.29σ1.0	2.49σ1.7	3038.06σ784.4	31.34σ7.0
−0.6	32.09σ13.0	−2.42σ1.7	**4.63** σ3.1	**3049.70** σ741.8	29.08σ8.3
−0.4	29.65σ13.2	**-1.82** σ1.5	3.20σ1.5	2967.62σ896.8	**32.44** σ15.3
−0.2	27.47σ11.2	−2.63σ0.9	2.73σ2.5	2761.86σ783.1	30.41σ9.3
0.0	22.73σ16.4	−2.42σ1.2	1.56σ2.0	2494.44σ791.0	28.70σ9.8
0.2	13.24σ11.7	−2.24σ1.6	−0.69σ2.0	2389.85σ649.7	16.98σ10.8
0.4	13.53σ7.1	−3.61σ0.7	0.92σ2.3	2477.75σ603.5	16.84σ11.7
0.6	14.03σ11.9	−2.78σ2.2	0.85σ1.5	2489.83σ682.9	14.46σ9.2
0.8	9.95σ10.0	−2.72σ1.5	1.20σ2.2	2391.18σ369.0	15.08σ12.4

FedPref algorithm does accomplish some useful collaboration leading to improvement of client results, but highly sparse solution spaces remain a challenge.

Deep-Sea Treasure. Sample results for the development of client similarity during training on the Deep-Sea Treasure environment with balanced and unbalanced preference assignment are shown in Fig. A.6 and Fig. A.7, respectively. In both figures, we observe that a grouping of clients becomes visible quite early in the learning process. Though this grouping is not perfect, it does largely correspond to those sets of clients that have been assigned the same preference. The flaws in the grouping process likely spring from an early clustering step, where preference similarities were not fully reflected in the respective model gradients.

MO-Halfcheetah. Uniquely for this environment, experiments were run with a lower number of clients, due to the high computing cost of solving this problem. For the equal

Table A.9: Numerical results for *topR* sensitivity analysis visualised in the main part of the paper. All configurations except those on the MO-HC environment were run with 10 different random seeds across 20 clients per run. Due to the higher computational cost of solving the MO-HC environment, experiments on this environment were restricted to 10 clients per run, also for 10 runs per configuration.

topR	MO-LL	DMC	DST	MO-HC	MO-LLc.
0.2	**33.18** σ11.1	−2.58 σ1.0	2.24 σ3.0	2909.41 σ636.1	30.51 σ9.3
0.4	30.04 σ11.1	−3.18 σ1.0	1.89 σ3.0	**3043.55** σ636.1	32.71 σ9.3
0.6	30.42 σ13.3	**-2.44** σ1.8	1.35 σ3.8	2995.96 σ893.1	32.04 σ12.5
0.8	31.17 σ11.7	−2.65 σ1.3	**3.76** σ4.0	2968.06 σ915.7	**33.39** σ11.3
1.0	30.84 σ10.4	−2.49 σ1.2	2.52 σ2.8	2967.83 σ714.8	32.03 σ10.7

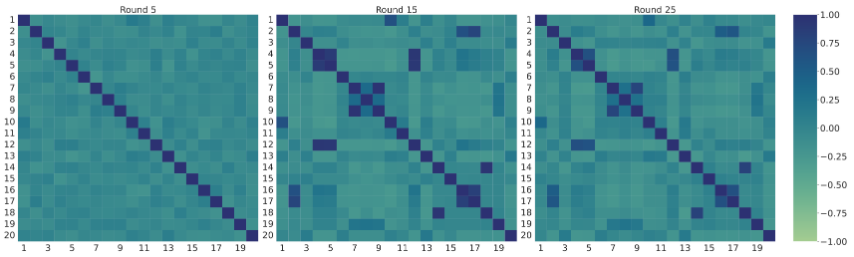


Figure A.4: Mutual client similarity at different stages during a single experimental run on the DMC environment, with balanced preference distribution. Left to right: client similarities after aggregation round 5, 15 and 25 of 38, respectively.

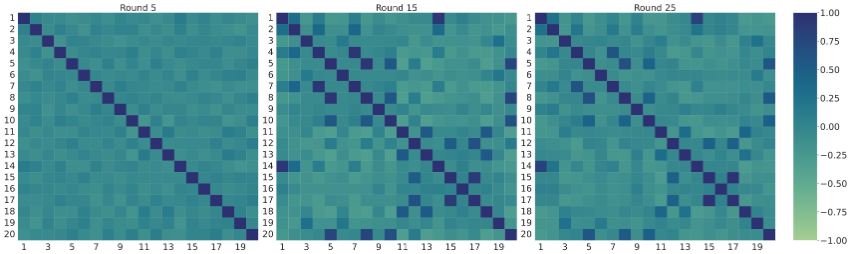


Figure A.5: Mutual client similarity at different stages during a single experimental run on the DMC environment, with unbalanced preference distribution. Left to right: client similarities after aggregation round 5, 15 and 25 of 38, respectively.

distribution, systems with 9 clients were constructed, with the same preference weights given to 3 clients each. For the unequal distribution scenario, 10 clients were run, with 1, 4, 3, and 2 clients receiving the same preferences, respectively. Sample results are shown in Fig. A.8 for the balanced distribution, and Fig. A.9 for the unbalanced distribution. For the results of the balanced distribution, we observe a fairly early tendency for dissimilarity between the clients, with some similarity grouping already apparent in the leftmost image, after five aggregation rounds. This early divergence between clients seems to lead in part to counter-intuitive clustering decisions, so that not all clients with the same similarity are grouped together. However, we note that, given the results seen in the main part of the

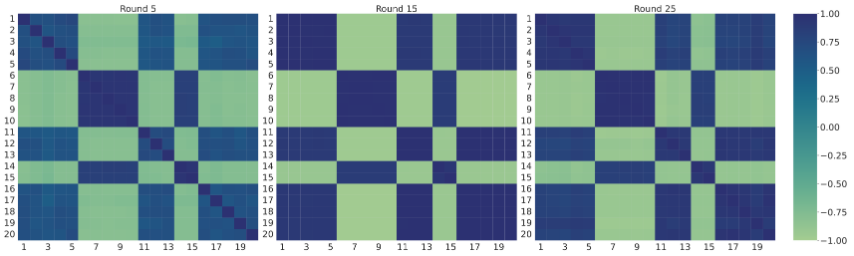


Figure A.6: Mutual client similarity at different stages during a single experimental run on the DST environment, with balanced preference distribution. Left to right: client similarities after aggregation round 5, 15 and 25 of 28, respectively.

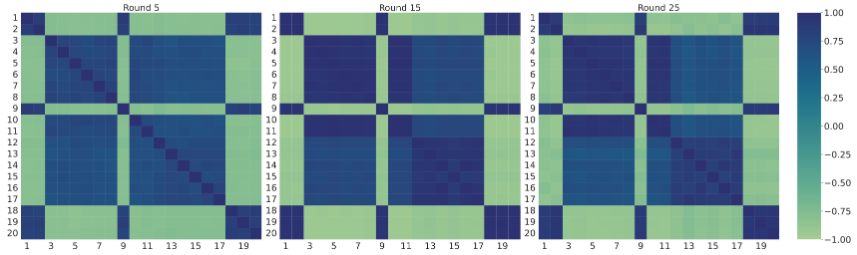


Figure A.7: Mutual client similarity at different stages during a single experimental run on the DST environment, with unbalanced preference distribution. Left to right: client similarities after aggregation round 5, 15 and 25 of 28, respectively.

paper, clients do appear to be able to learn together constructively. In the final image, we observe slightly less sharp dissimilarities between clients, indicating that further clustering has taken place, and most clients are likely entirely separated from the rest.
For the unbalanced preference distribution, we observe similar results, though interestingly the resulting clusters seem more appropriate to the underlying distribution structure. However, this difference could be a result of the particular experiment instances selected here for visualisation.

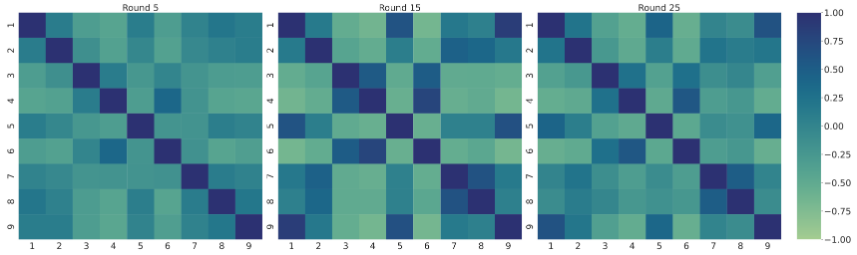


Figure A.8: Mutual client similarity at different stages during a single experimental run on the MO-HC environment, with balanced preference distribution. Left to right: client similarities after aggregation round 5, 15 and 25 of 30, respectively.

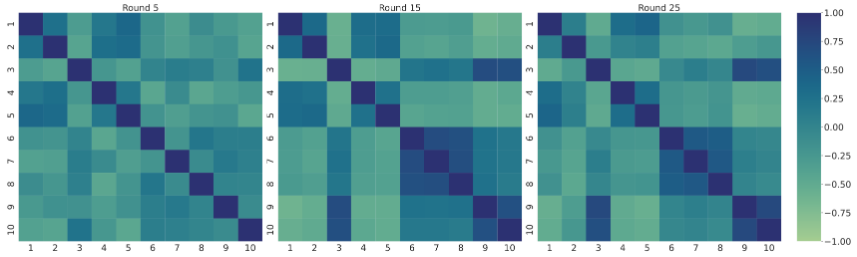


Figure A.9: Mutual client similarity at different stages during a single experimental run on the MO-HC environment, with unbalanced preference distribution. Left to right: client similarities after aggregation round 5, 15 and 25 of 30, respectively.

Continuous MO-Lunar Lander. Fig. A.10 and Fig. A.11 show sample similarity results for clients trained on the Continuous MO-Lunar Lander environment under balanced and unbalanced preference distributions, respectively. We observe that these results are

quite similar to those of the MO-LL environment, with near-perfect separation of clients into groups with the same preferences. For the visualised instance of the balanced distribution, these groups become visible almost immediately, already clearly recognisable after 5 aggregation rounds. Indeed, it appears that the clustering process fully separates clients with different preferences almost immediately, with no intermediate step with larger cluster groups discernible.
For the visualised instance of training on the unbalanced distribution, it takes markedly longer for clear similarity differences to become visible. This suggests that the clustering process takes longer to separate clients, either because they do indeed benefit from mutual collaboration for an extended time, or perhaps, as speculated earlier, because larger groups of clients with the same preferences dominate the intra-cluster training, skewing the cluster-mean convergence criterion. However, we note that in the final image, after 25 aggregation rounds, the separation of clients into clusters again matches the underlying distribution structure almost perfectly.

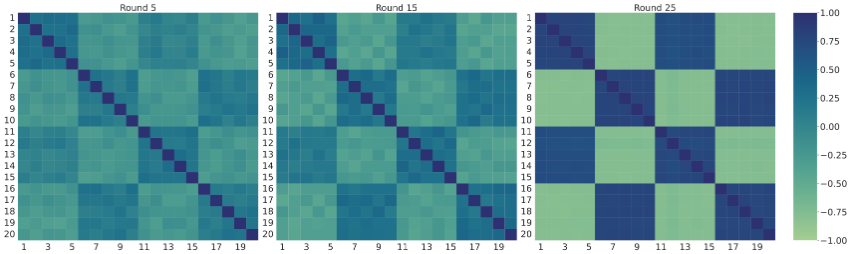


Figure A.10: Mutual client similarity at different stages during a single experimental run on the DST environment, with balanced preference distribution. Left to right: client similarities after aggregation round 5, 15 and 25 of 30, respectively.

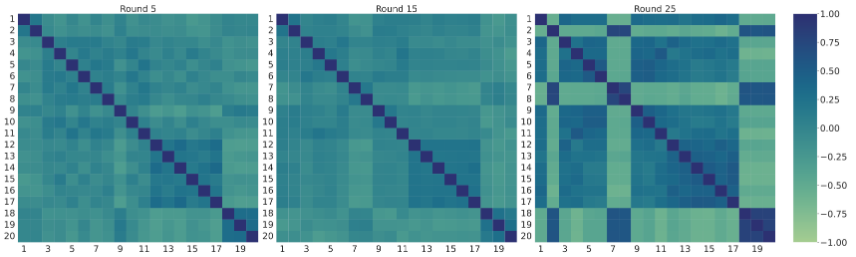


Figure A.11: Mutual client similarity at different stages during a single experimental run on the DST environment, with unbalanced preference distribution. Left to right: client similarities after aggregation round 5, 15 and 25 of 30, respectively.

A.2.2.3 | Investigating CFL clustering

In this subsection, we present a brief exploration of the clustering performance of the CFL algorithm on the experimental problems discussed in this paper. This investigation was sparked by the observation that the CFL algorithm often performed relatively similarly to the non-personalised FedAvg algorithm in our validation experiments. In Figures A.12 and A.13, we trace the evolution in the distribution of cluster sizes across aggregation rounds. In the interest of brevity, we include figures only for preferences generated under a

Dirichlet distribution; these are broadly representative of the corresponding results for the other distributions. We make two general observations:

- For some environments (MO-LL and DST), the clustering process is rarely triggered.
- If the clustering process is triggered, it appears to lead to very imbalanced clusters – often, single-client clusters are created.

Given that the clustering threshold parameters for CFL were selected following a hyperparameter search of parameter intervals recommended in [Sat19], we suspect that the first observation is explained by the second: perhaps the imbalanced clustering occurring here limits the performance of the algorithm sufficiently that the hyperparameter search leads to the selection of a threshold that is rarely triggered, avoiding clustering altogether. To explain the poor clustering performance itself, we propose two hypotheses. First, perhaps the size or training development of the RL models trained in these experiments impacts the success of the similarity metric computed on these models and so impacts the clustering process. Second, the greedy clustering algorithm proposed in [Sat19] may not favour the generation of balanced clusters.

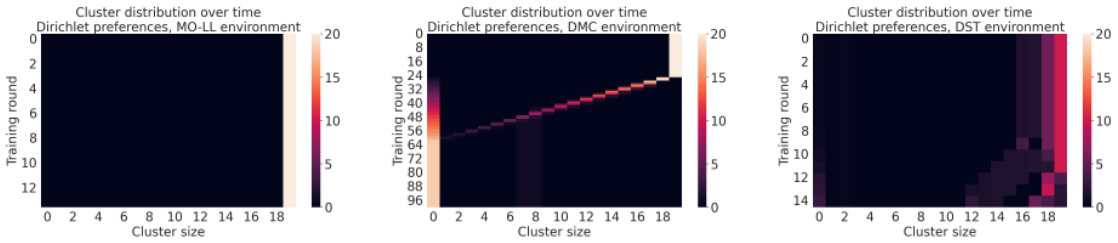


Figure A.12: Clustering behaviour of the CFL algorithm across different environments. Left to right: MO-LL, DMC, DST.

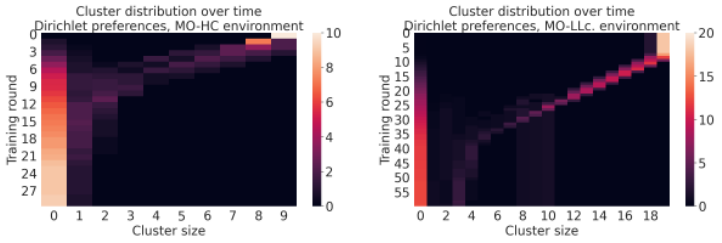


Figure A.13: Clustering behaviour of the CFL algorithm across different environments. Left to right: MO-HC, MO-LLcont.

A.3 | A new class of benchmarks for federated multi-objective learning

A.3.1 | Multi-MNIST experiments

The motivational experiment presented in Section 5.1 contrasts the results generated by FedAvg and the non-federated baseline when run with the same hyperparameters. Both variants were run on a minimal federated system of 2 clients, where the two clients are

assigned a contrasting preference distribution. Ten such configuration were generated, with preferences set to $[(0,1),(1,0)]$, $[(0.1,0.9),(0.9,0.1)]$, et cetera. Each preference configuration was run five times.

A.3.2 | Experimental configuration

Our model architecture for all experiments consists of a simple neural network with two hidden layers of size 64 and 32, respectively, using ReLU activation functions. For the output layer we use a Sigmoid activation function.

A.3.3 | Parameter tuning

We tune all algorithms by grid search, running each configuration for three runs. The tested parameter values are listed in Table A.10, with values that were ultimately selected shown in Table A.11 and Table A.12. The same three heterogeneous preference assignments were tested for each configuration across algorithms, with each set of preferences drawn uniformly at random from the weight simplex. In evaluating the results of the parameter search, we observe a Pareto front, with different configurations producing different trade-off solutions.

A.3.4 | Additional results

This section contains the experimental results that were discussed in the main paper, but could not be reported in detail. The remaining plots of Pareto fronts found by different algorithms on 10 clients are shown in Fig. A.14 and Fig. A.15 for experiments with homogeneous and heterogeneous preferences, respectively. The corresponding minimum and maximum values for each metric and experiment can be found in Tables A.13, A.14, A.15, and A.16.

In addition to the experiments presented in the main section, we have also carried out additional experiments scaled to federated systems of 50 clients. These results are visualised in Figs. A.16 and A.17 and Figs. A.18 and A.19 for homogeneous and heterogeneous preferences, respectively. The corresponding numerical hypervolume values may be found in Tables A.17 and A.20, with the minimum and maximum values reported in Tables A.18, A.19, A.21, and A.22.

A.3.5 | Practical remarks

In the main paper, we have noted the presence of statistical noise in client results. With multi-objective analysis in particular, outliers could distort the reported performance of algorithms, e.g. if identified as points on the Pareto front. Therefore, it may be useful to account for this noise in multi-objective analysis, e.g. by relaxing the strict Pareto front to one of rank k as defined in (Deb et al., 2002)[1], computed by removing the current non-dominated solutions from the solution set and computing the Pareto front of the remainder k times.

In this work, we successfully used a simple filtering rule to remove non-converged solutions, relying on the fact that perfect fairness is difficult to achieve for non-trivial

1 Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182-197

Table A.10: Complete list of parameter configurations tested during hyperparameter tuning of algorithms.

Algorithm	Parameter	Tested values	Comment
no comm	Learning rate	$5 \cdot 10^{-4}, 10^{-3}, 10^{-2}$	No federated parameters.
FedProx	Learning rate	$5 \cdot 10^{-4}, 10^{-3}, 10^{-2}$	
	Num. local iterations	$10, 25, 50$	
	Proximal term μ	$0, 0.01, 0.1$	$\mu = 0$ recovers standard FedAvg
	Finetuning rounds	$0, 1$	
CFL	Learning rate	$5 \cdot 10^{-4}, 10^{-3}, 10^{-2}$	
	Num. local iterations	$10, 25, 50$	
	Clustering threshold	$1, 2.5, 5, 7.5$	
	Patience	$1, 2$	Rounds below threshold before clustering triggered[a]
	Finetuning rounds	$0, 1$	
FedCMOO	Learning rate	$5 \cdot 10^{-4}, 10^{-3}, 10^{-2}$	
	Global learning rate	$1.0, 1.5, 2.0, 2.5$	
	Num. local iterations	$10, 25, 50$	
	Finetuning rounds	$0, 1$	
FedPref	Learning rate	$5 \cdot 10^{-4}, 10^{-3}, 10^{-2}$	
	Num. local iterations	$10, 25, 50$	
	Clustering threshold	$1, 2.5, 5, 7.5$	Relative change
	Patience	$1, 2$	Rounds below threshold before clustering triggered[a]
	Finetuning rounds	$0, 1$	

[a] Introduced by us to handle slow initial gradient ramp-up and noise introduced by client heterogeneity.

classifiers, and excluding all solutions with a fairness value greater than $1 - \epsilon$, with value of ϵ set in the range of 10^{-3}.

A common challenge in multi-objective optimization is an imbalance in the magnitude of different individual objective functions, as observed e.g. in Askin et al. [Ask24]. In settings such as this, where the potential values of the objective function are unbounded, an optimal mitigation strategy remains an open problem. Milojkovic et al. [Mil20] suggest normalising objective functions by the initial values obtained for each. We note this approach tends to favor fairness over accuracy objectives, given that most fairness metrics produce near-perfect scores for the uniform predictions generated by untrained models. However, in practice this normalization appears to work quite well for fairness problems, both in [Pad21] and our own experiments. Other normalization approaches are possible.

Finally, converging with a high preference for fairness can be difficult, as the perfect fairness of an untrained model does not give a sufficient impetus for a model to start learning. Following Padh et al. [Pad21], we mitigate this problem in our experiments by adding a small fraction of the accuracy loss for regularization, e.g. for DDP loss:

$$\text{Loss}_{DDP} = \widehat{DDP} + 0.1 \cdot \text{BCELogitsLoss}. \tag{A.1}$$

Table A.11: Parameter configurations selected for each algorithm and problem with the DEO fairness metric. Left to right: Adult dataset with gender as sensitive attribute, adult - race, Law School - gender, Law school - race, Default -gender.

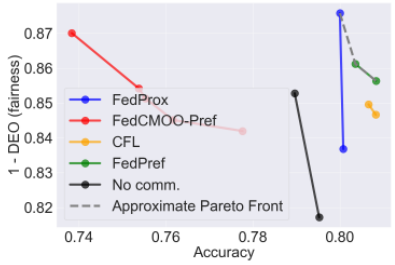
Algorithm	Parameter	AD - G	AD - R	LS - G	LS - R	DFT
no comm	Learning rate	$5 \cdot 10^{-4}$	10^{-3}	10^{-3}	10^{-3}	10^{-3}
FedProx	Learning rate	10^{-3}	10^{-3} 10^{-2}	10^{-2}	10^{-3}	
	Num. local iterations	50	50	25	25	50
	Proximal term μ	0	0.01	0	0.01	0
	Finetuning rounds	0	0	1	1	0
CFL	Learning rate	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}
	Num. local iterations	25	50	50	50	25
	Clustering threshold	7.5	5	7.5	5	7.5
	Patience	1	1	2	2	2
	Finetuning rounds	1	1	0	0	1
FedCMOO	Learning rate	10^{-2}	10^{-2}	10^{-3}	10^{-2}	10^{-2}
	Global learning rate	2.0	2.5	2.5	1.5	2.5
	Num. local iterations	10	50	50	10	25
	Finetuning rounds	0	0	0	1	0
FedPref	Learning rate	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-3}
	Num. local iterations	50	25	25	50	50
	Clustering threshold	7.5	2.5	1.0	1.0	1.0
	Patience	1	1	2	1	2
	Finetuning rounds	0	0	1	1	0

Table A.12: Parameter configurations selected for each algorithm and problem with the DDP fairness metric. Left to right: Adult dataset with gender as sensitive attribute, Adult - race, Law School - gender, Law school - race, Default -gender.

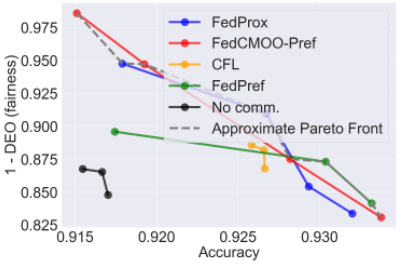
Algorithm	Parameter	AD - G	AD - R	LS - G	LS - R	DFT
no comm	Learning rate	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	10^{-3}	10^{-3}	$5 \cdot 10^{-4}$
FedProx	Learning rate	10^{-2}	10^{-2} 10^{-3}	10^{-3}	10^{-2}	
	Num. local iterations	25	25	10	10	10
	Proximal term μ	0	0.1	0	0.1	0.01
	Finetuning rounds	1	1	0	0	1
CFL	Learning rate	10^{-3}	10^{-3}	10^{-2}	10^{-2}	10^{-3}
	Num. local iterations	25	50	25	50	50
	Clustering threshold	1	1	5	7.5	2.5
	Patience	2	1	1	2	2
	Finetuning rounds	1	0	0	0	1
FedCMOO	Learning rate	10^{-2}	10^{-2}	10^{-2}	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
	Global learning rate	2.0	1.0	1.5	1.0	2.5
	Num. local iterations	50	25	50	10	25
	Finetuning rounds	0	0	0	0	0
FedPref	Learning rate	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-3}
	Num. local iterations	50	10	50	50	10
	Clustering threshold	2.5	1.0	5.0	5.0	7.5
	Patience	1	2	1	1	1
	Finetuning rounds	1	1	1	1	0

Table A.13: Range of global performance results for accuracy and DEO on 10 clients with homogeneous preferences. Only results from the algorithm-specific Pareto front are reported (see also Fig. 5.2 in the main section). All values scaled by 10^2. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

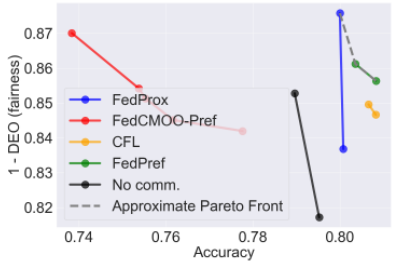
Data	FedProx	CFL	FedCMOO	FedPref	no comm
Sens. Attr.	Acc,DEO	Acc,DEO	Acc,DEO	Acc,DEO	Acc,DEO
Adult					
Gender min	(80.0,80.1)	(80.7,80.8)	(73.8,77.8)	(80.4,80.8)	(79.0,79.5)
max	(83.7,87.6)	(84.7,85.0)	(84.2,87.0)	(85.6,86.1)	(81.7,85.3)
Race min	(79.8,80.0)	(80.6,80.8)	(70.5,75.7)	(80.8,80.8)	(77.5,79.6)
max	(87.8,91.9)	(85.9,88.9)	(94.4,96.5)	(89.8,89.8)	(86.8,90.6)
Law School					
Gender min	(91.8,93.2)	(92.6,92.7)	(91.5,93.4)	(91.7,93.3)	(91.5,91.7)
max	(83.4,94.8)	(86.8,88.6)	(83.1,98.6)	(84.2,89.6)	(84.8,86.8)
Race min	(92.8,93.5)	(92.8,93.0)	(91.9,92.4)	(92.8,93.2)	(91.6,91.7)
max	(70.2,75.2)	(69.3,79.8)	(65.6,74.6)	(70.3,79.9)	(71.6,72.7)
Default					
Gender min	(71.4,73.0)	(68.5,70.1)	(64.3,73.6)	(72.1,74.4)	(70.0,70.4)
max	(91.6,96.8)	(94.6,96.2)	(94.0,98.6)	(93.5,96.9)	(93.4,96.5)



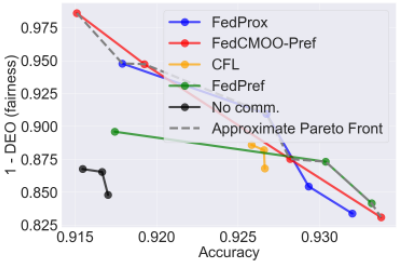
(a) Adult dataset with gender as the sensitive attribute and DEO fairness metric.



(b) Law School dataset with gender as the sensitive attribute and DEO fairness metric.

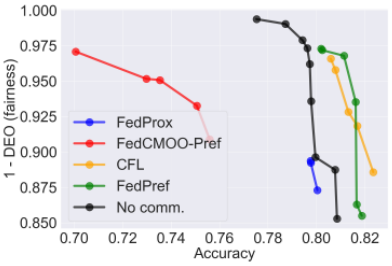


(c) Adult dataset with gender as the sensitive attribute and DDP fairness metric.

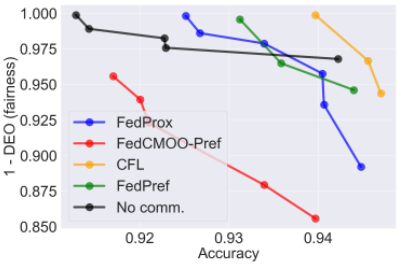


(d) Law School dataset with gender as the sensitive attribute and DDP fairness metric.

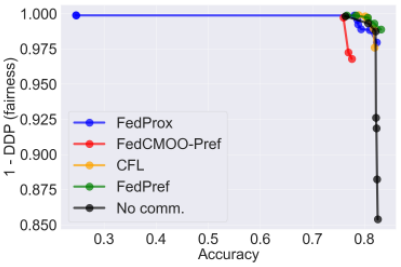
Figure A.14: Results of different algorithms on 10 clients on a selection of benchmark problems for accuracy and equality of opportunity (top row) and accuracy and demographic parity (bottom row). All clients were assigned the same preferences during a run, with 10 runs performed on preferences from $(0., 1.0)$ to $(0.9, 0.1)$, modified by steps of $(+0.1, -0.1)$. Each point represents the mean client output for a single run, with the Pareto fronts across all runs reported for each algorithm. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.



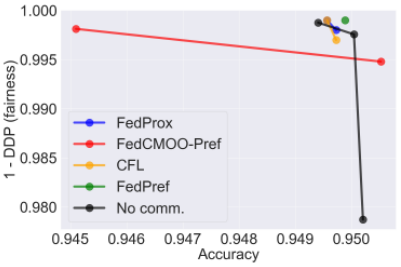
(a) Adult dataset with binarised race as the sensitive attribute and DEO fairness metric.



(b) Law School dataset with gender as the sensitive attribute and DEO metric.



(c) Adult dataset with binarised race as the sensitive attribute and DDP fairness metric.



(d) Law School dataset with binarised race as the sensitive attribute and DDP metric.

Figure A.15: Results of different algorithms on 10 clients on a selection of benchmark problems for accuracy and equality of opportunity (top row) and accuracy and demographic parity (bottom row). Clients were assigned heterogeneous preferences during each run, generated uniformly at random but the same across algorithms. Each point represents the output of a single client, with the Pareto fronts across all runs reported for each algorithm. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

Table A.14: Range of global performance results for accuracy and DDP (right) on 10 clients with homogeneous preferences. Only results from the algorithm-specific Pareto front are reported (see also Fig. 5.2 in the main section). All values scaled by 10^2. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

Data	FedProx	CFL	FedCMOO	FedPref	no comm
Sens. Attr.	Acc,DDP	Acc,DDP	Acc,DDP	Acc,DDP	Acc,DDP
Adult					
Gender min	(71.6,78.5)	(77.8,80.3)	(62.5,76.1)	(66.0,78.1)	(25.1,81.5)
max	(95.5,97.4)	(93.4,98.4)	(97.9,99.2)	(97.3,97.5)	(91.3,99.2)
Race min	(80.0,81.5)	(78.5,80.9)	(75.1,77.4)	(78.1,80.1)	(77.7,81.6)
max	(95.8,98.0)	(95.8,98.0)	(95.4,98.8)	(97.2,99.1)	(90.8,97.7)
Law School					
Gdr. min	(91.0,95.0)	(94.2,94.2)	(90.9,92.5)	(93.9,94.1)	(94.8,94.8)
max	(99.6,99.6)	(99.6, 99.6)	(99.2,99.5)	(99.6,99.6)	(99.5,99.5)
Race min	(93.1,93.1)	(94.4,94.8)	(94.5,94.9)	(93.8,94.6)	(94.4,94.8)
max	(97.9,97.9)	(99.0,99.6)	(99.2,99.6)	(97.9,98.1)	(98.9,99.1)
Default					
Gender min	(76.9,76.9)	(77.6,77.6)	(65.9,65.9)	(55.0,77.2)	(52.3,77.4)
max	(99.4,99.4)	(99.1,99.1)	(98.8,98.8)	(98.5,98.6)	(98.7,98.8)

Table A.15: Range of global performance results for accuracy and DEO with 10 clients on heterogeneous preferences. Only results from the algorithm-specific Pareto front are reported (see also Fig. 5.3 in the main section). All values scaled by 10^2. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

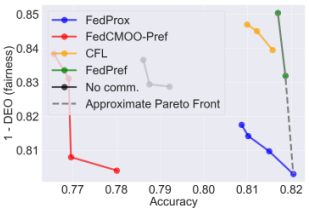
	FedProx	CFL	FedCMOO	FedPref	no comm
	Acc,DEO	Acc,DEO	Acc,DEO	Acc,DEO	Acc,DEO
Adult					
Gender min	(80.0, 80.1)	(80.7, 80.8)	(73.8, 77.8)	(80.4, 80.8)	(79.0, 79.5)
max	(83.7, 87.6)	(84.7, 85.0)	(84.2, 87.0)	(85.6, 86.1)	(81.7, 85.3)
Race min	(79.8, 80.0)	(80.6, 80.8)	(70.5, 75.7)	(80.8, 80.8)	(77.5, 79.6)
max	(87.8, 91.9)	(85.9, 88.9)	(94.4, 96.5)	(89.8, 89.8)	(86.8, 90.6)
Law School					
Gender min	(91.8, 93.2)	(92.6, 92.7)	(91.5, 93.4)	(91.7, 93.3)	(91.5, 91.7)
max	(83.4, 94.8)	(86.8, 88.6)	(83.1, 98.6)	(84.2, 89.6)	(84.8, 86.8)
Race min	(92.8, 93.5)	(92.8, 93.0)	(91.9, 92.4)	(92.8, 93.2)	(91.6, 91.7)
max	(70.2, 75.2)	(69.3, 79.8)	(65.6, 74.6)	(70.3, 79.9)	(71.6, 72.7)
Default					
Gender min	(71.4, 73.0)	(68.5, 70.1)	(64.3, 73.6)	(72.1, 74.4)	(70.0, 70.4)
max	(91.6, 96.8)	(94.6, 96.2)	(94.0, 98.6)	(93.5, 96.9)	(93.4, 96.5)

Table A.16: Range of global performance results for accuracy and DDP on 10 clients with heterogeneous preferences. Only results from the algorithm-specific Pareto front are reported (see also Fig. 5.3 in the main section). All values scaled by 10^2. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

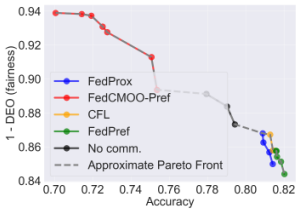
	FedProx	CFL	FedCMOO	FedPref	no comm
	Acc,DDP	Acc,DDP	Acc,DDP	Acc,DDP	Acc,DDP
Adult					
Gender min	(71.6, 78.5)	(77.8, 80.3)	(62.5, 76.1)	(66.0, 78.1)	(25.1, 81.5)
max	(95.5, 97.4)	(93.4, 98.4)	(97.9, 99.2)	(97.3, 97.5)	(91.3, 99.2)
Race min	(80.0, 81.5)	(78.5, 80.9)	(75.1, 77.4)	(78.1, 80.1)	(77.7, 81.6)
max	(95.8, 98.0)	(95.8, 98.0)	(95.4, 98.8)	(97.2, 99.1)	(90.8, 97.7)
Law School					
Gender min	(91.0, 95.0)	(94.2, 94.2)	(90.9, 92.5)	(93.9, 94.1)	(94.8, 94.8)
max	(99.6, 99.6)	(99.6, 99.6)	(99.2, 99.5)	(99.6, 99.6)	(99.5, 99.5)
Race min	(93.1, 93.1)	(94.4, 94.8)	(94.5, 94.9)	(93.8, 94.6)	(94.4, 94.8)
max	(97.9, 97.9)	(99.0, 99.6)	(99.2, 99.6)	(97.9, 98.1)	(98.9, 99.1)
Default					
Gender min	(76.9, 76.9)	(77.6, 77.6)	(65.9, 65.9)	(55.0, 77.2)	(52.3, 77.4)
max	(99.4, 99.4)	(99.1, 99.1)	(98.8, 98.8)	(98.5, 98.6)	(98.7, 98.8)

Table A.17: Hypervolumes of global performance results for accuracy and DEO (left) and accuracy and DDP (right) with 50 clients on homogeneous preferences. Higher is better (Fairness metrics are inverted, as in the results figures). Only results from the algorithm-specific Pareto front are reported (see also Fig. A.16 and Fig. A.17)

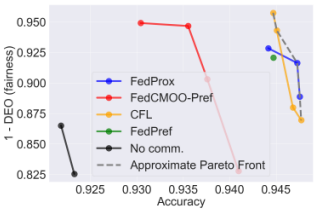
Data -	Accuracy - DEO					Accuracy - DDP				
Sens. attr.	FProx	CFL	FCMOO	FPref	no comm	FProx	CFL	FCMOO	FPref	no comm
Adult - G	0.671	0.691	0.654	**0.696**	0.663	0.776	**0.792**	0.767	0.769	0.781
Adult - R	0.707	0.706	0.707	0.703	**0.708**	0.770	**0.786**	0.740	0.779	0.759
Law - G	0.880	**0.907**	0.893	0.870	0.799	0.939	0.940	0.926	**0.941**	0.906
Law - R	0.723	0.664	0.702	0.718	**0.737**	**0.947**	0.945	0.944	0.932	0.904
Default - G	0.759	0.733	**0.799**	0.746	0.684	0.667	0.669	0.627	**0.684**	0.672



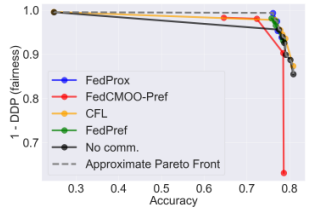
(a) Adult dataset with binarised gender as the sensitive attribute and DEO fairness metric.



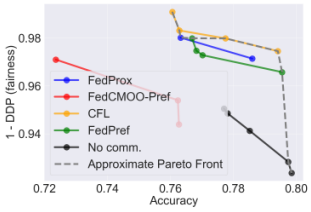
(b) Adult dataset with binarised race as the sensitive attribute and DEO fairness metric.



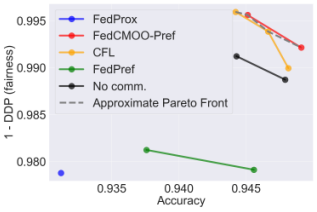
(c) Law School dataset with gender as the sensitive attribute and DEO metric.



(d) Adult dataset with binarised gender as the sensitive attribute and DDP fairness metric.

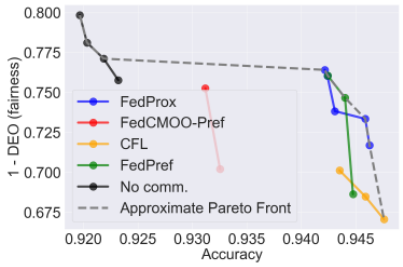


(e) Adult dataset with binarised race as the sensitive attribute and DDP fairness metric.

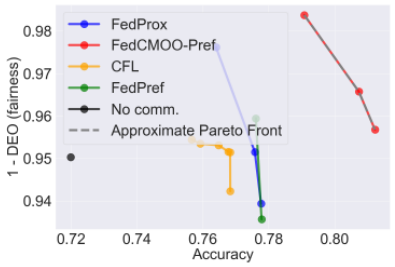


(f) Law School dataset with binarised race as the sensitive attribute and DDP metric.

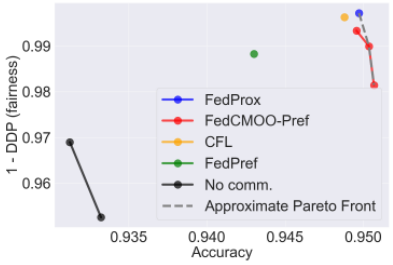
Figure A.16: Results of different algorithms on 50 clients on a selection of benchmark problems for accuracy and equality of opportunity (top row) and accuracy and demographic parity (bottom row). All clients were assigned the same preferences during a run, with 10 runs performed on preferences from $(0., 1.0)$ to $(0.9, 0.1)$, modified by steps of $(+0.1, -0.1)$. Each point represents the mean client output for a single run, with the Pareto fronts across all runs reported for each algorithm. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.



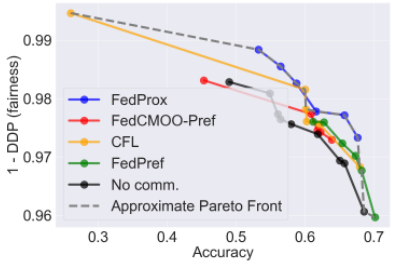
(a) Law school dataset with binarised race as the sensitive attribute and DEO fairness metric.



(b) Default dataset with gender as the sensitive attribute and DEO fairness metric.

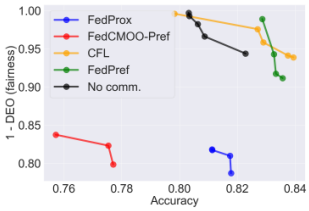


(c) Law School dataset with binarised race as the sensitive attribute and DDP fairness metric.

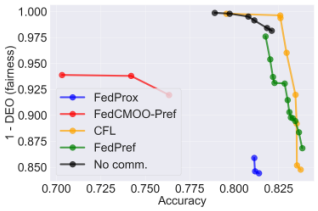


(d) Default dataset with gender as the sensitive attribute and DDP fairness metric.

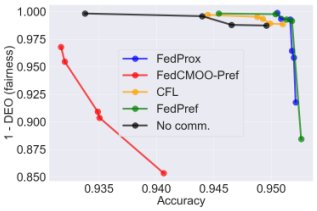
Figure A.17: Additional results of different algorithms on 50 clients on a selection of benchmark problems for accuracy and equality of opportunity (top row) and accuracy and demographic parity (bottom row). All clients were assigned the same preferences during a run, with 10 runs performed on preferences from $(0., 1.0)$ to $(0.9, 0.1)$, modified by steps of $(+0.1, -0.1)$. Each point represents the mean client output for a single run, with the Pareto fronts across all runs reported for each algorithm. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.



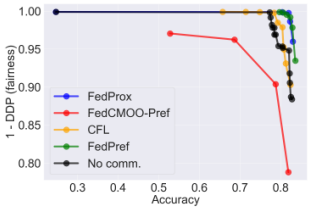
(a) Adult dataset with binarised gender as the sensitive attribute and DEO fairness metric.



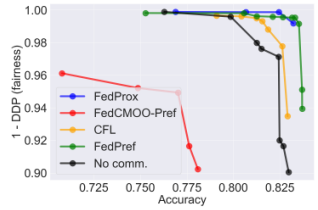
(b) Adult dataset with binarised race as the sensitive attribute and DEO fairness metric.



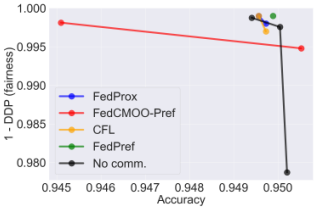
(c) Law School dataset with gender as the sensitive attribute and DEO metric.



(d) Adult dataset with binarised gender as the sensitive attribute and DDP fairness metric.

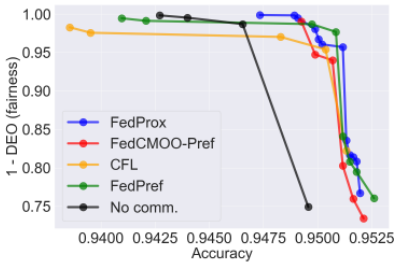


(e) Adult dataset with binarised race as the sensitive attribute and DDP fairness metric.

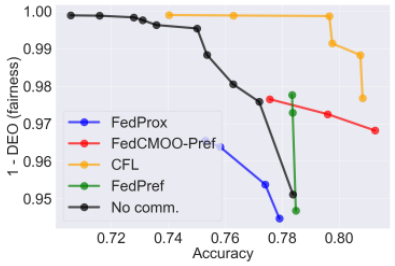


(f) Law School dataset with binarised race as the sensitive attribute and DDP metric.

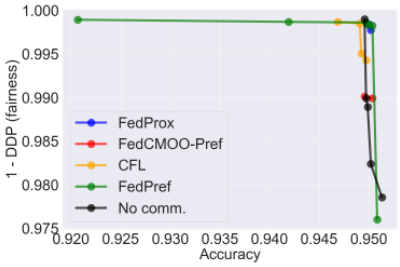
Figure A.18: Results of different algorithms on 50 clients on a selection of benchmark problems for accuracy and equality of opportunity (top row) and accuracy and demographic parity (bottom row). Clients were assigned heterogeneous preferences during each run, generated uniformly at random but the same across algorithms. Each point represents the output of a single client, with the Pareto fronts across all runs reported for each algorithm. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.



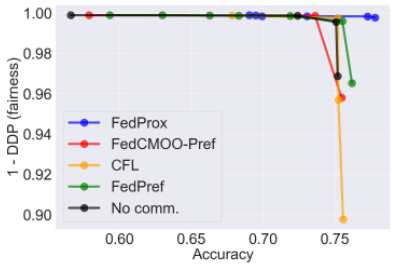
(a) Law school dataset with binarised race as the sensitive attribute and DEO fairness metric.



(b) Default dataset with gender as the sensitive attribute and DEO fairness metric.



(c) Law School dataset with binarised race as the sensitive attribute and DDP fairness metric.



(d) Default dataset with gender as the sensitive attribute and DDP fairness metric.

Figure A.19: Additional results of different algorithms on 50 clients on a selection of benchmark problems for accuracy and equality of opportunity (top row) and accuracy and demographic parity (bottom row). Clients were assigned heterogeneous preferences during each run, generated uniformly at random but the same across algorithms. Each point represents the output of a single client, with the Pareto fronts across all runs reported for each algorithm. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

Table A.18: Range of global performance results for accuracy and DEO on 50 clients with homogeneous preferences. Only results from the algorithm-specific Pareto front are reported (see also Fig. A.16 and Fig. A.17 in this appendix). All values scaled by 10^2. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

	FedProx	CFL	FedCMOO	FedPref	no comm
	Acc,DEO	Acc,DEO	Acc,DEO	Acc,DEO	Acc,DEO
Adult					
Gender min	(80.9, 82.0)	(81.0, 81.6)	(76.6, 78.0)	(81.7, 81.9)	(78.6, 79.2)
max	(80.3, 81.7)	(84.0, 84.7)	(80.4, 83.8)	(83.2, 85.0)	(82.9, 83.7)
Race min	(80.9, 81.4)	(81.3, 81.4)	(70.1, 75.4)	(81.6, 82.0)	(77.9, 79.4)
max	(85.0, 86.8)	(85.8, 86.7)	(89.4, 93.9)	(84.4, 85.8)	(87.3, 89.1)
Law School					
Gender min	(94.4, 94.8)	(94.5, 94.8)	(93.0, 94.1)	(94.5, 94.5)	(92.2, 92.3)
max	(88.9, 92.8)	(87.0, 95.7)	(82.8, 94.9)	(92.1, 92.1)	(82.5, 86.5)
Race min	(94.2, 94.6)	(94.4, 94.8)	(93.1, 93.3)	(94.2, 94.5)	(92.0, 92.3)
max	(71.7, 76.4)	(67.1, 70.1)	(70.2, 75.3)	(68.6, 76.1)	(75.8, 79.8)
Default					
Gender min	(76.4, 77.8)	(75.7, 76.8)	(79.1, 81.2)	(77.6, 77.8)	(72.0, 72.0)
max	(93.9, 97.6)	(94.2, 95.4)	(95.7, 98.4)	(93.6, 95.9)	(95.0, 95.0)

Table A.19: Range of global performance results for accuracy and DDP on 50 clients with homogeneous preferences. Only results from the algorithm-specific Pareto front are reported (see also Fig. A.16 and Fig. A.17 in this appendix). All values scaled by 10^2. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

	FedProx	CFL	FedCMOO	FedPref	no comm
	Acc,DDP	Acc,DDP	Acc,DDP	Acc,DDP	Acc,DDP
Adult					
Gender min	(76.1, 78.1)	(24.8, 80.8)	(64.6, 78.6)	(75.8, 78.4)	(24.9, 80.8)
max	(93.8, 99.4)	(87.3, 99.6)	(63.1, 98.3)	(93.1, 98.1)	(85.5, 99.5)
Race min	(76.3, 78.6)	(76.1, 79.4)	(72.4, 76.3)	(76.7, 79.5)	(77.7, 79.8)
max	(97.1, 98.0)	(97.5, 99.1)	(94.4, 97.1)	(96.6, 98.0)	(92.4, 95.1)
Law School					
Gender min	(90.7, 94.2)	(94.5, 94.5)	(92.9, 92.9)	(94.7, 94.7)	(91.1, 92.1)
max	(98.7, 99.6)	(99.5, 99.5)	(99.7, 99.7)	(99.5, 99.5)	(97.8, 98.3)
Race min	(95.0, 95.0)	(94.9, 94.9)	(95.0, 95.1)	(94.3, 94.3)	(93.1, 93.3)
max	(99.7, 99.7)	(99.6, 99.6)	(98.1, 99.3)	(98.8, 98.8)	(95.3, 96.9)
Default					
Gender min	(53.2, 67.6)	(26.0, 67.9)	(45.3, 63.9)	(61.2, 70.1)	(49.0, 68.6)
max	(97.3, 98.8)	(96.8, 99.5)	(97.3, 98.3)	(96.0, 97.6)	(96.1, 98.3)

Table A.20: Hypervolumes of global performance results for accuracy and DEO (left) and accuracy and DDP (right) with 50 clients on heterogeneous preferences. Higher is better (Fairness metrics are inverted, as in the results figures). Only results from the algorithm-specific Pareto front are reported (see also Fig. A.18 and Fig. A.19)

Data -	Accuracy - DEO					Accuracy - DDP				
Sens. attr.	FProx	CFL	FCMOO	FPref	no comm	FProx	CFL	FCMOO	FPref	no comm
Adult - G	0.669	**0.835**	0.650	0.826	0.820	0.829	0.821	0.781	**0.834**	0.824
Adult - R	0.699	**0.835**	0.716	0.817	0.820	0.831	0.826	0.749	**0.835**	0.828
Law - G	**0.951**	0.948	0.910	0.951	0.948	0.941	**0.950**	0.921	0.950	0.949
Law - R	**0.950**	0.935	0.943	0.947	0.947	0.949	0.948	0.941	0.950	**0.950**
Default - G	0.752	**0.807**	0.793	0.767	0.782	**0.777**	0.754	0.753	0.761	0.751

Table A.21: Range of global performance results for accuracy and DEO on 50 clients with heterogeneous preferences. Only results from the algorithm-specific Pareto front are reported (see also Fig. A.18 and Fig. A.19 in the appendix). All values scaled by 10^2. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

	FedProx	CFL	FedCMOO	FedPref	no comm
	Acc,DEO	Acc,DEO	Acc,DEO	Acc,DEO	Acc,DEO
Adult					
Gender min	(0.811, 0.787)	(0.798, 0.939)	(0.757, 0.799)	(0.829, 0.912)	(0.803, 0.944)
max	(0.818, 0.818)	(0.839, 0.996)	(0.777, 0.837)	(0.836, 0.989)	(0.823, 0.997)
Race min	(0.811, 0.844)	(0.795, 0.848)	(0.703, 0.920)	(0.818, 0.868)	(0.789, 0.982)
max	(0.814, 0.859)	(0.837, 0.998)	(0.763, 0.939)	(0.838, 0.976)	(0.821, 0.999)
Law School					
Gender min	(0.951, 0.918)	(0.944, 0.988)	(0.932, 0.853)	(0.945, 0.884)	(0.934, 0.987)
max	(0.952, 0.999)	(0.951, 0.997)	(0.941, 0.968)	(0.953, 0.998)	(0.950, 0.998)
Race min	(0.947, 0.767)	(0.939, 0.822)	(0.949, 0.734)	(0.941, 0.760)	(0.943, 0.749)
max	(0.952, 0.999)	(0.951, 0.983	(0.952, 0.990)	(0.953, 0.994)	(0.950, 0.998)
Default					
Gender min	(0.753, 0.945)	(0.740, 0.977)	(0.776, 0.968)	(0.783, 0.947)	(0.706, 0.951)
max	(0.779, 0.965)	(0.808, 0.999)	(0.813, 0.977)	(0.785, 0.978)	(0.784, 0.999)

Table A.22: Range of global performance results for accuracy and DDP on 50 clients with heterogeneous preferences. Only results from the algorithm-specific Pareto front are reported (see also Fig. A.18 and Fig. A.19 in this appendix). All values scaled by 10^2. All reported fairness metrics are inverted for ease of visualization, such that 1 corresponds to perfect fairness.

	FedProx	CFL	FedCMOO	FedPref	no comm
	Acc,DDP	Acc,DDP	Acc,DDP	Acc,DDP	Acc,DDP
Adult					
Gender min	(0.246, 0.960)	(0.657, 0.903)	(0.527, 0.788)	(0.797, 0.935)	(0.246, 0.884)
max	(0.830, 0.999)	(0.824, 0.999)	(0.819, 0.971)	(0.836, 0.999)	(0.828, 0.999)
Race min	(0.769, 0.992)	(0.791, 0.935)	(0.708, 0.902)	(0.753, 0.939)	(0.763, 0.900)
max	(0.832, 0.999)	(0.829, 0.997)	(0.781, 0.961)	(0.837, 0.998)	(0.830, 0.999)
Law School					
Gender min	(0.920, 0.993)	(0.894, 0.986)	(0.906, 0.997)	(0.898, 0.993)	(0.922, 0.989)
max	(0.943, 0.998)	(0.951, 0.999)	(0.922, 0.999)	(0.951, 0.999)	(0.950, 0.999)
Race min	(0.950, 0.998)	(0.947, 0.994)	(0.950, 0.990)	(0.921, 0.976)	(0.950, 0.979)
max	(0.950, 0.998)	(0.950, 0.999)	(0.950, 0.990)	(0.951, 0.999)	(0.951, 0.999)
Default					
Gender min	(0.690, 0.998)	(0.678, 0.898)	(0.579, 0.958)	(0.593, 0.965)	(0.566, 0.969)
max	(0.778, 0.999)	(0.756, 0.999)	(0.755, 0.999)	(0.762, 0.999)	(0.752, 0.999)