Learning Heterogeneous Performance-Fairness Trade-offs in Federated Learning

Rongguang Ye , **Ming Tang***
Department of Computer Science and Engineering and the Research Institute of Trustworthy Autonomous Systems at Southern University of Science and Technology, Shenzhen, China
yerg2023@mail.sustech.edu.cn, tangm3@sustech.edu.cn

Abstract

Recent methods leverage a hypernet to handle the performance-fairness trade-offs in federated learning. This hypernet maps the clients' preferences between model performance and fairness to preference-specifc models on the trade-off curve, known as local Pareto front. However, existing methods typically adopt a uniform preference sampling distribution to train the hypernet across clients, neglecting the inherent heterogeneity of their local Pareto fronts. Meanwhile, from the perspective of generalization, they do not consider the gap between local and global Pareto fronts on the global dataset. To address these limitations, we propose HetPFL to effectively learn both local and global Pareto fronts. HetPFL comprises Preference Sampling Adaptation (PSA) and Preference-aware Hypernet Fusion (PHF). PSA adaptively determines the optimal preference sampling distribution for each client to accommodate heterogeneous local Pareto fronts. While PHF performs preference-aware fusion of clients' hypernets to ensure the performance of the global Pareto front. We prove that HetPFL converges linearly with respect to the number of rounds, under weaker assumptions than existing methods. Extensive experiments on four datasets show that HetPFL significantly outperforms seven baselines in terms of the quality of learned local and global Pareto fronts.

1 Introduction

Federated Learning (FL) [McMahan *et al.*, 2017] is an emerging machine learning paradigm that designed to train neural network models using data silos while preserving data privacy. In recent years, FL has achieved remarkable success across various domains, including healthcare [Rieke *et al.*, 2020], fintech [Imteaj and Amini, 2022], and the Internet of Things (IoT) [Nguyen *et al.*, 2021]. As FL continues to develop, the issue of group fairness has become an increasingly significant focus. Specifically, there are two primary types of group fairness in FL: client-based fairness [Li *et al.*, 2019; Lyu *et al.*, 2020; Wang *et al.*, 2021] and group-based fairness [Yue *et al.*, 2023; Deng *et al.*, 2020]. Client-based fairness aims to minimize the variance in model performance across clients while preserving the overall model performance. Our work focuses on group-based fairness, which ensures that a model performs equitably across different demographic subgroups (e.g., male or female) within each local dataset [Kamishima *et al.*, 2012; Roh *et al.*, 2020].

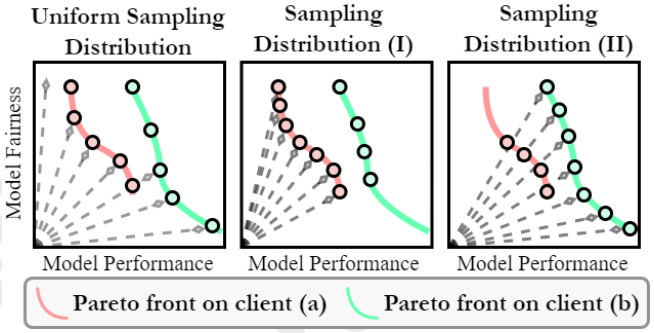

Figure 1: The impact of different sampling distributions under two clients. The dotted vectors represent preferences for model's performance and fairness. The pink and green points are loss vectors of the model after evaluation on the local dataset on client (a) and client (b), respectively. A uniform preference sampling distribution cannot achieve the best result of learning local Pareto fronts based on Lemma 1. Instead, sampling distribution (I) is suitable for client (a), and sampling distribution (II) is suitable for client (b).

Recent studies have focused on improving the group-based fairness of FL by proposing data sampling strategies and designing new optimization objectives. In terms of data sampling strategies, FedFB [Zeng *et al.*, 2021] adjusts the sampling probabilities of subgroup samples during training, increasing the probability of underperforming groups to achieve group fairness. Meanwhile, FairFed [Ezzeldin *et al.*, 2023] adopts a fairness-aware model aggregation scheme. Regarding the design of optimization objectives, LFT+FedAvg [Zeng *et al.*, 2023] incorporates group fairness as a constraint during local training. FAIRFATE [Salazar *et al.*, 2023] introduces a linear combination of model performance and group fairness as its objective function. In fact, a trade-off exists between model performance and model fairness, meaning that improving fair-

*Corresponding Author.

ness often comes at the cost of performance. With respect to efficiency, learning the entire performance-fairness trade-off curve (i.e., the Pareto front) for each client offers a more flexible scheme. Moreover, in terms of generalization, it is crucial to consider the quality of the global Pareto front on the global dataset. Recent studies [Lin *et al.*, 2022; Ye *et al.*, 2025] have introduced hypernets to learn the local Pareto front on the local dataset by modeling the mapping from a predefined preference distribution to preference-specific models. However, these methods still exhibit limitations in efficiency and generalization within the FL context:

- **Efficiency**: As shown in Fig. 1, the heterogeneity of data in FL results in differences in the positions of local Pareto fronts across clients. Each local Pareto front has its own optimal preference sampling distribution. However, prior approaches assume a uniform preference sampling distribution across clients, which is inefficient for learning local Pareto fronts.
- **Generalization**: Achieving both optimal local and global Pareto fronts presents inherent conflicts. Prior approaches primarily focus on improving the local Pareto front while neglecting the global Pareto front.

Addressing these limitations presents several challenges. In terms of efficiency, determining the optimal preference sampling distribution for each client is non-trivial, as the position of each client's local Pareto front (ground truth) is initially unknown. Regarding generalization, aggregating clients' models to construct the optimal global model is difficult, as it requires identifying the strengths of each client's model for different preferences while maintaining data privacy.

In this paper, we propose HetPFL to efficiently learn both local and global Pareto fronts. HetPFL consists of Preference Sampling Adaptation (PSA) and Preference-aware Hypernet Fusion (PHF). PSA dynamically adjusts the preference sampling distribution by introducing data-driven HyperVolume Contribution (HVC), which quantifies each preference's contribution to the learned Pareto front. We then jointly optimize the preference sampling distribution based on HVC and the client's model as a bi-level optimization problem to enhance local Pareto front learning efficiency. To improve the global Pareto front, PHF considers a preference-aware hypernet aggregation at the server by identifying the capability of each client's hypernet for various preferences.

The primary contributions of this work are as follows:

- We propose a HetPFL framework that efficiently learns the heterogeneous local Pareto fronts across clients using PSA, while simultaneously achieving a high-quality global Pareto front through PHF;
- We analyze the convergence rate of HetPFL within the FL system and establish an error convergence rate of order $\mathcal{O}\left(\frac{1}{t}\right)$. This result is particularly challenging to derive due to the interdependence of the different components in the FL system;
- Extensive experiments on four datasets demonstrate that HetPFL outperforms the best-performing baseline, achieving approximately 1.75% and 5.5% improvements in the quality of the learned local and global Pareto fronts, respectively.

2 Problem Formulation

Let $\boldsymbol{x} \in \mathcal{X}$, $y \in \mathcal{Y}$ denote features and label, respectively. The features in $\boldsymbol{x} \triangleq (\boldsymbol{a}, \boldsymbol{b})$ contain sensitive features $\boldsymbol{a}$ (e.g., gender, race) and non-sensitive features $\boldsymbol{b}$. Suppose there are K clients, and each client $k \in [K]$ has a local dataset, denoted by $\mathcal{D}_k$. The classifier $f_{\boldsymbol{\theta}_k}$ with learnable parameters $\boldsymbol{\theta}_k$ of client k outputs a prediction $f_{\boldsymbol{\theta}_k}(\boldsymbol{x})$ from an input data $\boldsymbol{x}$.

According to [Zeng *et al.*, 2021], we define loss functions for model performance and fairness, respectively. Usually, model performance is characterized using cross-entropy loss

$$\ell_{CE}(\boldsymbol{x}, y \,|\, f_{\boldsymbol{\theta}_k}) = -[y \log(f_{\boldsymbol{\theta}_k}(\boldsymbol{x})) + (1-y)\log(1 - f_{\boldsymbol{\theta}_k}(\boldsymbol{x}))]. \quad (1)$$

We use following loss function to quantify model fairness:

$$\ell_F(\boldsymbol{x}, y \mid f_{\boldsymbol{\theta}_k}) = \left[(\boldsymbol{a} - \bar{\boldsymbol{a}}_k)(f_{\boldsymbol{\theta}_k}(\boldsymbol{x}) - \bar{f}_{\boldsymbol{\theta}_k}(\boldsymbol{x}))\right], \quad (2)$$

where $\bar{\boldsymbol{a}}_k$ and $\bar{f}_{\boldsymbol{\theta}_k}(\boldsymbol{x})$ represent the average values of $\boldsymbol{a}$ and $f_{\boldsymbol{\theta}_k}(\boldsymbol{x})$ over $\mathcal{D}_k$, respectively. ℓ_F measures the correlation between sensitive features and model predictions. When the model prediction exhibits a stronger correlation with sensitive features, the value of ℓ_F rises, signaling a reduced model fairness.

The trade-off between ℓ_{CE} and ℓ_F is quantified using a preference vector $\boldsymbol{\lambda} \in \Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}_+^2 \mid \sum_{i=1}^2 \lambda_i = 1\}$. Following [Ye *et al.*, 2025], we introduce a hypernet $h_{\boldsymbol{\beta}_k} : \mathbb{R}^{|\boldsymbol{\lambda}|} \to \mathbb{R}^{|\boldsymbol{\theta}_k|}$ with learnable parameters $\boldsymbol{\beta}_k$, which maps a preference vector $\boldsymbol{\lambda}$ to a preference-specific model $\boldsymbol{\theta}_k = h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda})$. We aim at optimizing $\boldsymbol{\beta}_k$ to improve model performance and fairness (i.e., reducing the losses in Eqs. (1) and (2)), for which we can define a *weighted Tchebycheff scalar* loss [Miettinen, 1999] for each preference vector $\boldsymbol{\lambda}$:

$$\min_{\boldsymbol{\beta}_k} g_{\text{tch}}(\boldsymbol{x}, y, h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda}) \mid \boldsymbol{\lambda}) = \max_{j \in \{CE, F\}} \left\{ \frac{\ell_j(\boldsymbol{x}, y \mid h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda}))}{\lambda_j} \right\}. \quad (3)$$

Eq. (3) satisfies the following Lemma [Miettinen, 1999]:

Lemma 1 (Preference Alignment)**.** *Given a preference vector $\boldsymbol{\lambda}$, a preference-specific model $h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda})$ is weakly Pareto optimal to the problem (3) if and only if $h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda})$ is optimal for problem (3).*

Lemma 1 guarantees that when $h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda})$ is optimal for problem (3), the loss vector (ℓ_{CE}, ℓ_F) of $h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda})$ on dataset $\mathcal{D}_k$ aligns exactly with the direction of the preference vector and lies on the Pareto front, as shown by the points in Fig. 1.

We consider optimizing Eq. (3) over the preference distribution Λ_k of each client k, and define the following goal for the ***local Pareto front*** learning of client k.

$$\min_{\boldsymbol{\beta}_k} \mathbb{E}_{\boldsymbol{\lambda} \sim \Lambda_k} \mathbb{E}_{(\boldsymbol{x}, y) \in \mathcal{D}_k} g_{\text{tch}}(\boldsymbol{x}, y, h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda}) \mid \boldsymbol{\lambda}), \quad (4)$$

where Λ_k is unknown in advance and depends on the position of the Pareto front of client k. Preference vector $\boldsymbol{\lambda}$ is sampled from Λ_k. Once Eq. (4) is completed, the hypernet can receive all possible preference vectors as inputs, generating a corresponding set of preference-specific models $\{\boldsymbol{\theta}_k = h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda}) \mid \boldsymbol{\lambda} \sim \Lambda_m\}$. This model set is then evaluated on the local dataset $\mathcal{D}_k$, and the evaluation results collectively form the entire local Pareto front.

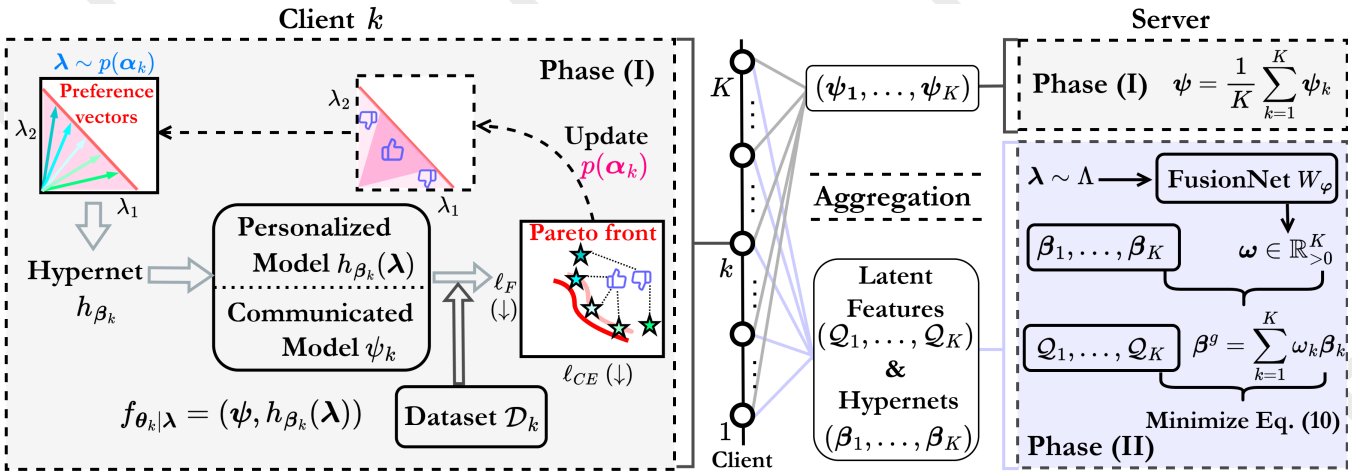

Figure 2: HetPFL framework.

Similarly, the goal for the ***global Pareto front*** is to generate an aggregated hypernet $\boldsymbol{\beta}^g = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{\beta}_k$, which minimizes the $g_{\text{tch}}(\cdot)$ over the global dataset:

$$\min_{\boldsymbol{\beta}^g} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{\lambda}\sim\Lambda} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_k} g_{\text{tch}}(\boldsymbol{x}, y, h_{\boldsymbol{\beta}^g}(\boldsymbol{\lambda}) \mid \boldsymbol{\lambda}), \quad (5)$$

where Λ can be an arbitrary distribution or defined as a combination of $\Lambda_1, ..., \Lambda_K$.

Our goal is to optimize Eq. (4) for all clients while simultaneously optimizing Eq. (5). The primary challenge in Eq. (4) arises from the distinctiveness of each client's local Pareto front. This implies the necessity of approximating preference distribution Λ_k for each client k to effectively learn the local Pareto fronts. For Eq. (5), the optimization objectives for local and global Pareto fronts are inherently conflicting, preventing simultaneous optimality for both.

3 Methodology

This section presents our HetPFL framework. We provide an overview and introduce the two main components, PSA and PHF, of HetPFL in Sections 3.1–3.3. Sections 3.4–3.5 describe the optimization procedure of HetPFL and analyze its convergence properties.

3.1 Overview

Fig. 2 shows our proposed HetPFL. The foundational components of HetPFL include communicated model $\boldsymbol{\psi}$, hypernet $h_{\boldsymbol{\beta}_k}$, preference sampling distribution $p(\boldsymbol{\alpha}_k)$, and FusionNet $W_{\boldsymbol{\varphi}}$. HetPFL can be divided into two phases. Phase (I) focuses on efficiently learning the local Pareto fronts for all clients, while Phase (II) aims to learn the global Pareto front.

Phase (I)

In this phase, we aim to optimize the hypernet, sampling distribution, and the communicated model. The communicated model $\boldsymbol{\psi}$ transforms the features into d-dimensional latent features and is periodically aggregated at the server, as in FL. The hypernet $h_{\boldsymbol{\beta}_k}$ is kept locally at the client in Phase (I). Its role is to map an arbitrary preference vector $\boldsymbol{\lambda}$ into a preference-specific model $h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda})$, and then $h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda})$ transforms the d-dimensional latent features to label. For each client k, we denote $f_{\boldsymbol{\theta}_k|\boldsymbol{\lambda}} = (\boldsymbol{\psi}, h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda}))$.

Previous works simply set preference sampling distribution $p(\boldsymbol{\alpha}_k)$ to be uniform across all clients. In contrast, we consider jointly optimizing the hypernet $h_{\boldsymbol{\beta}_k}$ and $p(\boldsymbol{\alpha}_k)$:

$$\min_{\boldsymbol{\beta}_k, \boldsymbol{\alpha}_k} \mathbb{E}_{\boldsymbol{\lambda}\sim p(\boldsymbol{\alpha}_k)} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_k} g_{\text{tch}}(\boldsymbol{x}, y, f_{\boldsymbol{\theta}_k|\boldsymbol{\lambda}}), \quad (6)$$

where $\boldsymbol{\alpha}_k$ are the parameters of $p(\boldsymbol{\alpha}_k)$. HetPFL improves the efficiency of learning local Pareto fronts by identifying a suitable $p(\boldsymbol{\alpha}_k)$ for each client over the preference space. Meanwhile, the communicated model optimized to improve the model performance of all clients:

$$\min_{\boldsymbol{\psi}} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_k} \left[\ell_{CE}(\boldsymbol{x}, y, f_{\boldsymbol{\theta}_k|\widetilde{\boldsymbol{\lambda}}}) \right], \quad (7)$$

where $f_{\boldsymbol{\theta}_k|\widetilde{\boldsymbol{\lambda}}} = (\boldsymbol{\psi}, h_{\boldsymbol{\beta}_k}(\widetilde{\boldsymbol{\lambda}}))$ and $\widetilde{\boldsymbol{\lambda}}$ is a predefined preference vector. The details of update steps for Eqs. (6) and (7) are provided in Section 3.4.

Phase (II)

Unlike previous works that overlook the global Pareto front, Phase (II) focuses on addressing this gap. After the final round, clients first transmit their hypernets and latent features, $\mathcal{Q}_k = \boldsymbol{\psi}(\boldsymbol{x}) \mid (\boldsymbol{x}, y) \in \mathcal{D}k$, to the server. The server then uses FusionNet, $W\boldsymbol{\varphi}$, with parameters $\boldsymbol{\varphi}$ to learn effective aggregation strategies for these hypernets, tailoring the aggregation process to different preference vectors. Notably, transmitting only the latent features of the dataset is a common practice to mitigate privacy concerns [Thapa *et al.*, 2022].

Given this framework, two key questions remain to be addressed. First, how can $p(\boldsymbol{\alpha}_k)$ be determined in Phase (I) to enable efficient learning of local Pareto fronts? Second, how to optimize FusionNet in Phase (II)? These questions will be explored in the following two subsections.

3.2 Preference Sampling Adaptation

In Phase (I), we propose Preference Sampling Adaptation (PSA) to determine $p(\boldsymbol{\alpha}_k)$. Determining $p(\boldsymbol{\alpha}_k)$ involves optimizing the quality of the sampled preference vectors during training. This process is non-trivial and unfolds in two steps: first, evaluating the quality of the sampled preference vectors without a true Pareto front (ground truth); and second, integrating the optimization of $p(\boldsymbol{\alpha}_k)$ into the hypernet's training.

For the first step, we introduce a data-driven Hypervolume Contribution (HVC) indicator to assess the quality of sampled preference vectors. Despite the absence of a true local Pareto front, it allows for quantifying each preference vector's contribution based on the training losses.

Definition 1 (HVC)**.** *Given a reference point* $\boldsymbol{r}$*. Let* $\mathcal{S}(\boldsymbol{\lambda}, \boldsymbol{r}) = \left\{\boldsymbol{q} \in \mathbb{R}^2 \mid \boldsymbol{\ell}(\boldsymbol{x}, y \mid f_{\boldsymbol{\theta}_k|\boldsymbol{\lambda}}) \leq \boldsymbol{q} \text{ and } \boldsymbol{q} \leq \boldsymbol{r}\right\}$*, and the hypervolume of a set of* N *preference vectors* $\Lambda_{\boldsymbol{\alpha}_k} = \{\boldsymbol{\lambda}^1, ..., \boldsymbol{\lambda}^N \mid \boldsymbol{\lambda}^i \sim p(\boldsymbol{\alpha}_k)\}$ *is*

$$\mathcal{H}_{\boldsymbol{r}}\left(\Lambda_{\boldsymbol{\alpha}_k}\right) = \mathcal{L}\left(\bigcup_{\boldsymbol{\lambda} \in \Lambda_{\boldsymbol{\alpha}_k}} \mathcal{S}(\boldsymbol{\lambda}, \boldsymbol{r})\right),$$

where $\mathcal{L}(\cdot)$ *denotes the Lebesgue measure and* $\boldsymbol{q}$ *represents any point in the gray area in the left figure of Fig. 3. The HVC of* $f_{\boldsymbol{\theta}|\boldsymbol{\lambda}^i}$ *for the set* $\Lambda_{\boldsymbol{\alpha}_k}$ *is the difference between* $\mathcal{H}_{\boldsymbol{r}}(\Lambda_{\boldsymbol{\alpha}_k})$ *and* $\mathcal{H}_{\boldsymbol{r}}(\Lambda_{\boldsymbol{\alpha}_k} \setminus \boldsymbol{\lambda}^i)$*, as follows:*

$$\mathcal{HC}_{\boldsymbol{r}}(\boldsymbol{\lambda}^i \mid \Lambda_{\boldsymbol{\alpha}_k}) = \mathcal{H}_{\boldsymbol{r}}(\Lambda_{\boldsymbol{\alpha}_k}) - \mathcal{H}_{\boldsymbol{r}}(\Lambda_{\boldsymbol{\alpha}_k} \setminus \boldsymbol{\lambda}^i).$$

Fig. 3 shows that the HVC of $f_{\boldsymbol{\theta}_k|\boldsymbol{\lambda}^i}$ is the difference between the HV of the full set of five models and that of the set excluding $f_{\boldsymbol{\theta}_k|\boldsymbol{\lambda}^i}$. The larger $\mathcal{HC}_{\boldsymbol{r}}(\boldsymbol{\lambda}^i \mid \Lambda_{\boldsymbol{\alpha}_k})$, the greater the contribution of $\boldsymbol{\lambda}^i$, indicating a higher quality of $\boldsymbol{\lambda}^i$.

We then move on to the second step. Based on HVC, we propose the following bi-level optimization objective to alternately optimize the hypernet and the preference sampling distribution of client k:

$$\min_{\boldsymbol{\alpha}_k} \mathbb{E}_{\boldsymbol{\lambda} \sim p(\boldsymbol{\alpha}_k)} \mathbb{E}_{(\boldsymbol{x}, y) \in \mathcal{D}_k}[-\mathcal{HC}_{\boldsymbol{r}}\left(\boldsymbol{\lambda} \mid \Lambda_{\boldsymbol{\alpha}_k}\right)], \tag{8}$$

$$\text{s.t. } \boldsymbol{\beta}_k = \arg\min_{\boldsymbol{\beta}_k} \mathbb{E}_{\boldsymbol{\lambda} \sim p(\boldsymbol{\alpha}_k)} \mathbb{E}_{(\boldsymbol{x}, y) \in \mathcal{D}_k}[g_{\text{tch}}(\boldsymbol{x}, y, f_{\boldsymbol{\theta}_k|\boldsymbol{\lambda}})], \tag{9}$$

where $f_{\boldsymbol{\theta}_k|\boldsymbol{\lambda}} = \left(\boldsymbol{\psi}_k, h_{\boldsymbol{\beta}_k}(\boldsymbol{\lambda})\right)$. The bi-level optimization first optimizes the hypernet as in Eq. (9). Then, the sampling distribution is further refined based on the solution of Eq. (9). As shown in Fig. 2, it makes the sampling distribution at the next iteration more beneficial to local Pareto front learning.

3.3 Preference-aware Hypernet Fusion

In Phase (II), we propose a preference-aware hypernet fusion (PHF) method. The intuition behind PHF is that each client's hypernet excels at specific preference vectors. Thus, for any given preference vector, PHF learns the preference-aware aggregation weight so that hypernets specializing in that vector are given higher weight, thereby improving the quality of the global Pareto front. As shown in Fig. 2, we introduce a FusionNet $W_{\boldsymbol{\varphi}} : \mathbb{R}^{|\boldsymbol{\lambda}|} \to \mathbb{R}^K_{>0}$ with parameters $\boldsymbol{\varphi}$ that learns a mapping from any preference vector $\boldsymbol{\lambda}$ to a fusion weight $\boldsymbol{\omega} = W_{\boldsymbol{\varphi}}(\boldsymbol{\lambda}) \in \mathbb{R}^K$. Based on the fusion weight, the hypernets from all clients are linearly combined to form a global hypernet $\boldsymbol{\beta}^g = W_{\boldsymbol{\varphi}}(\boldsymbol{\lambda}) \cdot [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K]$, where operation $\cdot$ is the vector inner product. This process is optimized through

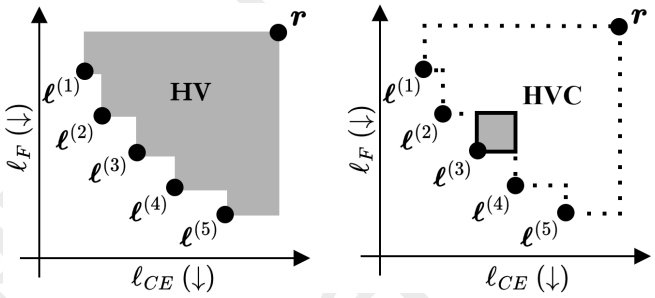

Figure 3: An illustration of the HV and HVC of the loss vectors $\boldsymbol{\ell}^{(i)}$ produced by evaluating a model set $\{f_{\boldsymbol{\theta}_k|\boldsymbol{\lambda}^i}\}_{i=1}^5$ corresponding to five input preference vectors.

$$\min_{\boldsymbol{\varphi}} \mathbb{E}_{\boldsymbol{\lambda} \sim \Lambda}\left[\frac{1}{K}\sum_{k=1}^{K} g_{\text{tch}}(\mathcal{Q}_k, \mathcal{Y}_k, f_{\boldsymbol{\theta}_k^g|\boldsymbol{\lambda}})\right], \tag{10}$$

where $f_{\boldsymbol{\theta}_k^g|\boldsymbol{\lambda}} = (\boldsymbol{\psi}, h_{\boldsymbol{\beta}^g}(\boldsymbol{\lambda}))$, and $\mathcal{Y}_k$ represents labels on dataset $\mathcal{D}_k$. Recalling Lemma 1, the global hypernet $f_{\boldsymbol{\theta}_k^g|\boldsymbol{\lambda}}$ aggregated by FusionNet is optimized towards the direction where the Pareto front intersects with $\boldsymbol{\lambda}$. Once Eq. (10) is solved, the mapping from preference vectors to fusion weights during inference is highly efficient.

3.4 Algorithm: HetPFL

In this subsection, we present HetPFL algorithm to optimizing four components including communicated model $\boldsymbol{\psi}$, hypernet $h_{\boldsymbol{\beta}_k}$, preference sampling distribution $p(\boldsymbol{\alpha}_k)$, and FusionNet $W_{\boldsymbol{\varphi}}$. At the beginning of Phase (I), each client downloads the global communicated model $\boldsymbol{\theta}_k^0$ from the server.

In round t, the communicated model on client k is updated through τ_c steps of gradient descent with a learning rate of η_t:

$$\boldsymbol{\psi}^t \leftarrow \boldsymbol{\psi}^t - \eta_t \mathbb{E}_{(\boldsymbol{x}, y) \in \mathcal{D}_k} \nabla_{\boldsymbol{\psi}}\left[\boldsymbol{\ell}_{CE}(\boldsymbol{x}, y \mid f_{\boldsymbol{\theta}_k|\widetilde{\boldsymbol{\lambda}}})\right]. \tag{11}$$

To balance model performance and fairness, we set $\widetilde{\boldsymbol{\lambda}}$ to $(\frac{1}{2}, \frac{1}{2})$ in Eq. (11). Then, we proceed to optimize the hypernet and the preference sampling distribution (in Eqs. (9) and (8)). Note that lower-level problem (Eq. (9)) is a stochastic optimization problem, which is challenging to solve directly due to the expectation over preference distribution involving infinite possible values. To address this, we approximate the expectation term using Monte Carlo sampling and then solve the Eq. (9) with τ_p steps of gradient descent

$$\boldsymbol{\beta}_k^t \leftarrow \boldsymbol{\beta}_k^t - \frac{\eta_t}{N} \mathbb{E}_{(\boldsymbol{x}, y) \in \mathcal{D}_k} \sum_{v=1}^{N} \nabla_{\boldsymbol{\beta}_k} g_{\text{tch}}(\boldsymbol{x}, y, f_{\boldsymbol{\theta}_k|\boldsymbol{\lambda}^v} | \boldsymbol{\lambda}^v), \tag{12}$$

where η_t denotes the learning rate, $\boldsymbol{\lambda}^v$ is a sampled preference vector and N is the number of sampled preference vectors.

Solving the upper-level problem in Eq. (8) relies on computing the HVC gradient, given by $g_{\boldsymbol{\alpha}_k} = \nabla_{\boldsymbol{\alpha}_k} \mathbb{E}_{\boldsymbol{\lambda} \in \Lambda_{\boldsymbol{\alpha}_k}} \mathbb{E}_{(\boldsymbol{x}, y) \in \mathcal{D}_k}[-\mathcal{HC}_{\boldsymbol{r}}(\boldsymbol{\lambda} \mid \Lambda_{\boldsymbol{\alpha}_k})]$. However, since this gradient is sometimes non-differentiable, we employ

Natural Evolution Strategies (NES) [Salimans *et al.*, 2017], which yield gradient estimation $\hat{g}_{\boldsymbol{\alpha}_k}$ for $g_{\boldsymbol{\alpha}_k}$:

$$\hat{g}_{\boldsymbol{\alpha}_k} \approx \mathbb{E}_{\boldsymbol{\lambda}\in\Lambda_{\boldsymbol{\alpha}_k}}\left[-\mathcal{HC}_{\boldsymbol{r}}(\boldsymbol{\lambda}|\Lambda_{\boldsymbol{\alpha}_k})\nabla_{\boldsymbol{\alpha_k}}\log p(\boldsymbol{\lambda}|\boldsymbol{\alpha}_k)\right], \quad (13)$$

where $\Lambda_{\boldsymbol{\alpha}_k}$ is the set of N preference vectors collected in Eq. (12). This gradient computation method only requires the preference sampling distribution $p(\boldsymbol{\alpha}_k)$ to be differentiable, without the need for HVC function to be differentiable. Based on Eq. (13), to optimize Eq. (8), $\boldsymbol{\alpha}_k$ is updated using the gradient $\hat{g}_{\boldsymbol{\alpha}}$ by performing τ_p steps of gradient descent:

$$\boldsymbol{\alpha}_k^t \leftarrow \boldsymbol{\alpha}_k^t - \kappa_t \hat{g}_{\boldsymbol{\alpha}_k}, \quad (14)$$

where κ_t is the learning rate. After round t is completed, all clients transmit their communicated models $\boldsymbol{\psi}_k, k \in [K]$, to the server. The server updates the communicated model by performing an averaging aggregation

$$\boldsymbol{\psi}_k^{t+1} = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{\psi}_k^t. \quad (15)$$

Subsequently, each client initializes the communicated model as the aggregated model for round $t+1$.

Upon completing a total of T rounds, we proceed to Phase (II), where the optimization of $\boldsymbol{\varphi}^t$ is updated by

$$\boldsymbol{\varphi}^t \leftarrow \boldsymbol{\varphi}^t - \eta_t \frac{1}{N}\frac{1}{K}\sum_{v=1}^{N}\sum_{k=1}^{K}\nabla_{\boldsymbol{\varphi}} g_{\mathrm{tch}}(\mathcal{Q}_k, \mathcal{Y}, f_{\boldsymbol{\theta}_k^g|\boldsymbol{\lambda}} \mid \boldsymbol{\lambda}), \quad (16)$$

where η_t is the learning rate.

3.5 Theoretical Analysis

In this subsection, we theoretically analyze the convergence of HetPFL algorithm. Our proof process is structured in two main steps. Firstly, we establish an upper bound of the communicated model at any given round t. Next, we provide the upper bound of the hypernet at any given round t.

To simplify the notation, we represent the expressions of Eqs. (8) and (9) as $g_{\mathrm{hvc}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k) = \mathbb{E}_{\boldsymbol{\lambda}\sim p(\boldsymbol{\alpha}_k)}\mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_k}[-\mathcal{HC}_{\boldsymbol{r}}(\boldsymbol{\lambda} \mid \Lambda_{\boldsymbol{\alpha}_k})]$ and $g_{\mathrm{tch}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k) = \mathbb{E}_{\boldsymbol{\lambda}\sim p(\boldsymbol{\alpha}_k)}\mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_k}[g_{\mathrm{tch}}(\boldsymbol{x}, y, f_{\boldsymbol{\theta}_{k|\boldsymbol{\lambda}}} \mid \boldsymbol{\lambda})]$, respectively.

Based on [Hong *et al.*, 2023], we make the following assumptions.

Assumption 1. *$\nabla_{\boldsymbol{\beta}} g_{\mathrm{tch}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)$, $\nabla^2_{\boldsymbol{\alpha\beta}} g_{\mathrm{tch}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)$, $\nabla^2_{\boldsymbol{\beta\beta}} g_{\mathrm{tch}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)$, $\nabla_{\boldsymbol{\alpha}} g_{\mathrm{hvc}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)$, and $\nabla_{\boldsymbol{\beta}} g_{\mathrm{hvc}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)$ are Lipschitz continuous in $\boldsymbol{\beta}_k$ with respective Lipschitz constants L_{t1}, L_{t2}, L_{t3}, L_{h1} and L_{h2}.*

Assumption 2. *$\nabla^2_{\boldsymbol{\alpha\beta}} g_{\mathrm{tch}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)$, $\nabla^2_{\boldsymbol{\beta\beta}} g_{\mathrm{tch}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)$, $\nabla_{\boldsymbol{\beta}} g_{\mathrm{hvc}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)$ is Lipschitz continuous in $\boldsymbol{\alpha}_k$ with respective Lipschitz constants L_{t4}, L_{t5} and L_{h3}.*

Assumption 3. *$g_{\mathrm{tch}}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)$ is μ_1-strongly convex in $\boldsymbol{\beta}_k$, and $g_{\mathrm{tch}}(\boldsymbol{\beta}_k, \boldsymbol{\alpha}_k^*)$ is μ_2-strongly convex in $\boldsymbol{\beta}_k$, where $\boldsymbol{\alpha}_k^*$ is optimal sampling distribution for client k.*

Assumption 4. *The expectation of stochastic gradients is always bounded. That is, $||\nabla^2_{\boldsymbol{\alpha\beta}} g_{tch}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)|| \leq G_1$, $||\nabla_{\boldsymbol{\alpha}} g_{hvc}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)|| \leq G_2$, and $||\nabla_{\boldsymbol{\beta}} g_{tch}(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)|| \leq G_3$.*

Let $\Delta^t_{\boldsymbol{\beta}_k} \triangleq \mathbb{E}\left[||\boldsymbol{\beta}_k^t - \boldsymbol{\beta}^*_{k|\boldsymbol{\alpha}_k^{t-1}}||^2\right]$ denote the error between the hypernet at round t and the optimal hypernet $\boldsymbol{\beta}^*_{k|\boldsymbol{\alpha}_k^{t-1}}$ given the sampling distribution $p(\boldsymbol{\alpha}_k^{t-1})$ in round $t-1$. Let $\Delta^t_{\boldsymbol{\psi}_k} \triangleq \mathbb{E}\left[||\boldsymbol{\psi}_k^t - \boldsymbol{\psi}_k^*||^2\right]$ denote the error between the communicated model at round t and the optimal communicated model $\boldsymbol{\psi}_k^*$. The communicated model has following upper bound.

Lemma 2 (Convergence of the Communicated Model [Collins *et al.*, 2021])**.** *If the communicated model $\boldsymbol{\psi}$ is optimized by FedAvg [McMahan* et al.*, 2017] and given a constant $\zeta > 0$, then $\boldsymbol{\psi}$ converges to the optimal communicated model $\boldsymbol{\psi}^*$ at a linear rate:*

$$\Delta^t_{\boldsymbol{\psi}_k} \leq (1-\eta\zeta)^{t/2}\,\Delta^0_{\boldsymbol{\psi}_k}, \quad (17)$$

with a probability at least $1 - te^{-100\min(|\boldsymbol{x}|^2\log(|K|),d)}$.

Lemma 2 shows that the error convergence rate is $\mathcal{O}(\frac{1}{t})$. Under weaker assumptions compared to [Ye *et al.*, 2025] (i.e., without requiring the initial convergence error of the hypernet to be a constant multiple of the communicated model), we establish the following upper bound for hypernet.

Theorem 1 (Convergence of the Hypernet)**.** *Under Assumptions 1-4 and Lemma 2, the upper bound of hypernet is*

$$\begin{aligned}\Delta^{t+1}_{\boldsymbol{\beta}_k} \leq\ & (\frac{3}{4})^{\tau_p t}\Delta^0_{\boldsymbol{\beta}_k} + z_1(1-\eta_t\zeta)^{t/4}\sqrt{\Delta^0_{\boldsymbol{\psi}_k}} \\ & + z_2(1-\eta_t\zeta)^{t/2}\Delta^0_{\boldsymbol{\psi}_k} + \frac{\sigma_1^2\mu_1 + c_1^2L_{q1}^2 + G_3^2\mu_1}{\mu_1^3} \\ & + 2\eta_t L_{t1}(1-\eta_t\zeta)^{t/4}\sqrt{\Delta^0_{\boldsymbol{\psi}_k}\Delta^0_{\boldsymbol{\beta}_k}},\end{aligned} \quad (18)$$

where z_1, z_2 are constants, and $\Delta^t_{\boldsymbol{\alpha}_0} = \mathbb{E}\left[||\boldsymbol{\alpha}_k^0 - \boldsymbol{\alpha}_k^*||^2\right]$. Theorem 1 guarantees an optimization error of order $\mathcal{O}((\frac{3}{4})^{\tau_p t} + (1-\eta_t\zeta)^{t/4} + (1-\eta_t\zeta)^{t/2} + \frac{\sigma_1^2\mu_1 + c_1^2L_{q1}^2 + G_3^2\mu_1}{\mu_1^3})$. When $t \to +\infty$, $\Delta^{t+1}_{\boldsymbol{\beta}_k}$ converges to $\frac{\sigma_1^2\mu_1 + c_1^2L_{q1}^2 + G_3^2\mu_1}{\mu_1^3}$. Due to $(1-\eta_t\zeta)^{t/4}$ being the dominant term in the error convergence rate, the overall error convergence rate is $\mathcal{O}(\frac{1}{t})$.

4 Experiments

4.1 Experimental Settings

Datasets. Four widely-used datasets are employed to evaluate the performance of HetPFL, including a SYNTHETIC [Zeng *et al.*, 2021], COMPAS [Barenstein, 2019], BANK [Moro *et al.*, 2014], and ADULT [Dua *et al.*, 2017].

Baselines. We compare HetPFL with seven state-of-the-art methods, including two for addressing local fairness (LFT+Ensemble and LFT+Fedavg [Zeng *et al.*, 2023]), three for global fairness (Agnosticfair [Du *et al.*, 2021], FairFed [Ezzeldin *et al.*, 2023] and FedFB [Zeng *et al.*, 2021]), one for both local and global fairness ([Makhija *et al.*, 2024]), and one for learning performance-fairness local Pareto fronts (PraFFL [Ye *et al.*, 2025]). PraFFL is the most closely related work to ours in learning Pareto fronts.

Metrics. Based on [Ezzeldin *et al.*, 2023], we use the model's error rate to quantify its performance and the DP disparity [Feldman *et al.*, 2015] to measure its fairness, where a smaller

Method	SYNTHETIC		COMPAS		BANK		ADULT	
	Local HV	Global HV	Local HV	Global HV	Local HV	Global HV	Local HV	Global HV
LFT+Ensemble	0.425	0.479	0.514	0.555	0.890	0.881	0.760	0.764
LFT+Fedavg	0.700	0.468	0.505	0.514	0.891	0.138	0.765	0.501
Agnosticfair	0.492	0.537	0.499	0.550	0.887	0.880	<u>0.780</u>	<u>0.783</u>
FairFed	0.339	0.367	0.418	0.434	0.889	0.878	0.267	0.270
FedFB	0.567	0.608	0.505	0.517	0.893	0.883	0.759	0.763
EquiFL	0.642	0.604	0.564	0.526	0.892	0.882	0.761	0.764
PraFFL	<u>0.800</u>	<u>0.716</u>	<u>0.599</u>	<u>0.613</u>	<u>0.901</u>	<u>0.895</u>	0.766	0.750
HetPFL (Ours)	**0.830**	**0.827**	**0.623**	**0.626**	**0.904**	**0.898**	**0.783**	**0.846**

Table 1: Averaged performance comparison of different methods across four datasets over three runs. The best results are highlighted in bold, while the second-best results are underlined.

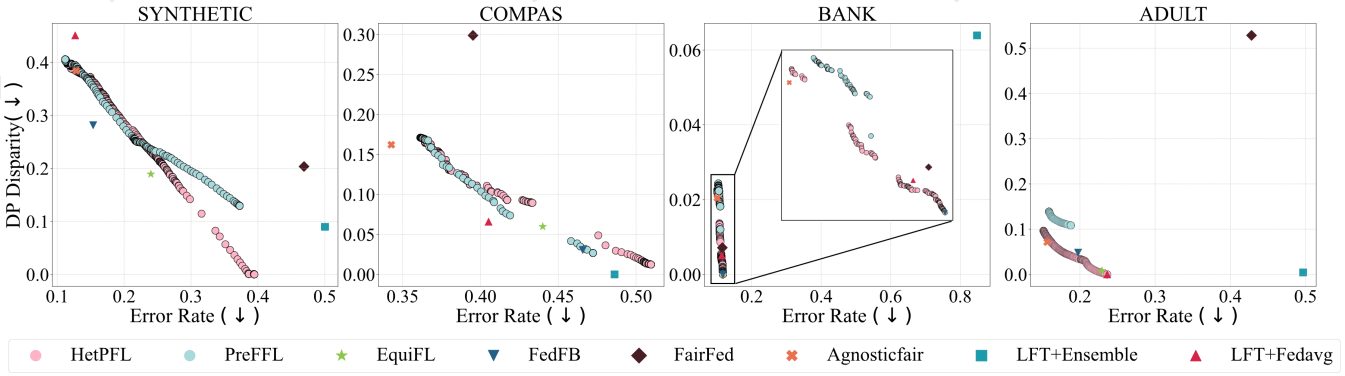

Figure 4: Comparison of global Pareto front obtained by our HetPFL algorithm and baselines on four datasets. A Pareto front closer to the bottom-left corner indicates better performance.

DP disparity indicates a fairer model. Additionally, the hypervolume (HV) [Zitzler and Thiele, 1999] is employed to evaluate the quality of the learned Pareto front. We mainly report local HV on local datasets and global HV on global datasets.

Hyperparameters. Since PraFFL and HetPFL have the ability to generate any number of models during inference, we set them to generate 1,000 preference-specific models each for evaluation. Our implementation is available at `https://github.com/rG223/HetPFL`.

4.2 Experimental Results

Main Results. We draw two key conclusions based on Table 1: (I) Methods capable of learning the Pareto front, such as PraFFL and HetPFL, outperform those that do not in most cases, in terms of both local HV and global HV. Fig. 4 shows that PraFFL and HetPFL are capable of generating more models that caters to large-scale preferences; (II) PraFFL focuses primarily on learning local Pareto fronts, it neglects the heterogeneity of Pareto fronts across clients and fails to ensure the quality of the global Pareto front, leading suboptimal on most datasets. In comparison, HetPFL outperforms PraFFL in terms of local HV and global HV across four datasets, achieving varying degrees of improvement. Fig. 4 indicates that HetPFL outperforms PraFFL on SYNTHETIC, BANK, and ADULT datasets. However, on the COMPAS, the Pareto front splits into two segments with a 3% disconnection in terms of DP disparity and error rate. HetPFL's unimodal sampling distribution prioritizes the tail regions' benefits while overlooking the middle sections compared to PraFFL.

Convergence Results. Fig. 5 shows the convergence comparison of PraFFL and HetPFL on the client local validation set in each round. HetPFL shows consistently faster convergence compared to the PraFFL, validating the effectiveness of our proposed preference sampling adaptation method.

The Impact of Data Heterogeneity. Table 2 demonstrates that HetPFL consistently achieves the best performance in both local HV and global HV across all levels of data heterogeneity compared to seven baselines. Notably, the following observations emerge: (I) **Comprehensiveness**: First five baselines in Table 2 fail to learn the entire Pareto front and struggle with high data heterogeneity. Their performance deteriorates as heterogeneity increases. HetPFL not only learns the entire Pareto front but also handles high heterogeneity effectively; (II) **Scalability**: When compared with personalized FL methods such as EquiFL and PraFFL, our proposed HetPFL excels in handling high data heterogeneity in both local and global datasets. EquiFL and PraFFL are better at handling high heterogeneity on local datasets, but they fail to address heterogeneity effectively on the global dataset.

The Impact of the Number of Clients. We analyze Table 3 from following two aspects: (I) **Comprehensiveness**:

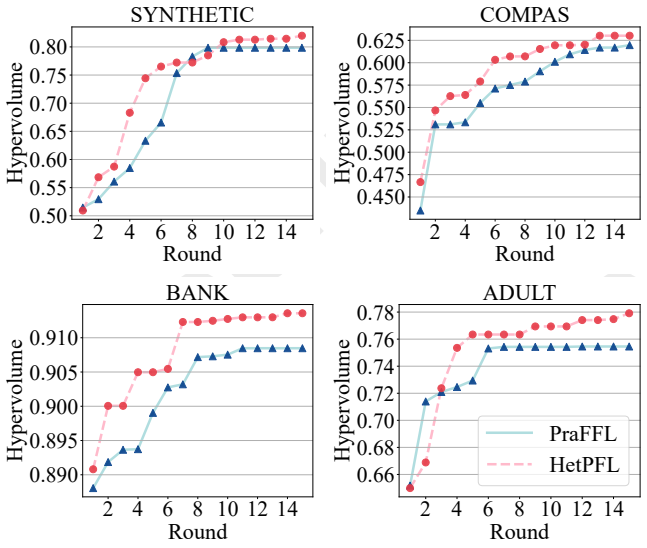

Figure 5: Convergence of HetPFL compared with PraFFL.

Method	Local HV			Global HV		
	Heterogeneity Param.			Heterogeneity Param.		
	0.1	5	1000	0.1	5	1000
LFT+Ensemble	0.311	0.481	0.463	0.487	0.487	0.475
LFT+Fedavg	0.593	0.692	0.701	0.468	0.468	0.468
Agnosticfair	0.417	0.535	0.526	0.537	0.537	0.537
FairFed	0.392	0.349	0.410	0.429	0.400	0.403
FedFB	0.518	0.602	0.597	0.616	0.615	0.607
EquiFL	0.734	0.626	0.615	0.584	0.644	0.643
PraFFL	0.802	0.781	0.789	0.707	0.744	0.768
HetPFL	**0.808**	**0.806**	**0.810**	**0.820**	**0.791**	**0.817**

Table 2: Performance comparison across different heterogeneity levels on the SYNTHETIC dataset. The smaller the heterogeneity parameter, the greater the data heterogeneity.

The first six methods in Table 3, which lack the capability to learn the Pareto front, exhibit limited performance in both small and large-scale client scenarios, with both local HV and global HV consistently below 0.7. In contrast, HetPFL not only learns the Pareto front but also achieves the best performance across all scenarios, with both local HV and global HV exceeding 0.78; (II) **Scalability**: Compared to PraFFL, HetPFL demonstrates clear advantages in large-scale client scenarios. Notably, PraFFL tends to collapse in learning the global Pareto front under large-scale settings, whereas HetPFL consistently achieves the best results in both local HV and global HV, regarding different client scales.

Ablation Study. Table 4 reveals two key observations: (I) PSA enhances the learning ability of the local Pareto front. On the SYNTHETIC dataset, the local HV improves from 0.80 to 0.83 with PSA. Similar improvements in local HV can be observed across the other three datasets. (II) PHF enhances the performance of the global Pareto front. Without PSA, PHF achieves an approximately 8% improvement in global

Method	Local HV			Global HV		
	Number of Clients			Number of Clients		
	10	100	300	10	100	300
LFT+Ensemble	0.475	0.464	0.535	0.463	0.536	0.470
LFT+Fedavg	0.664	0.472	0.605	0.460	0.589	0.469
Agnosticfair	0.548	0.557	0.547	0.551	0.635	0.553
FairFed	0.414	0.326	0.413	0.372	0.381	0.353
FedFB	0.610	0.599	0.562	0.613	0.667	0.581
EquiFL	0.564	0.602	0.620	0.613	0.573	0.607
PraFFL	0.803	0.716	0.789	0.807	0.578	0.621
HetPFL	**0.808**	**0.785**	**0.848**	**0.808**	**0.814**	**0.813**

Table 3: Performance comparison of different methods across the number of clients on SYNTHETIC dataset.

Dataset	PSA	PHF	Local HV	Global HV
SYNTHETIC	×	×	0.800	0.719
	×	✓	0.800	0.793
	✓	×	0.830	0.825
	✓	✓	**0.830**	**0.827**
COMPAS	×	×	0.599	0.613
	×	✓	0.599	**0.639**
	✓	×	0.623	0.624
	✓	✓	**0.623**	0.626
BANK	×	×	0.901	0.895
	×	✓	0.901	0.896
	✓	×	0.904	0.886
	✓	✓	**0.904**	**0.898**
ADULT	×	×	0.766	0.750
	×	✓	0.766	0.846
	✓	×	0.783	0.813
	✓	✓	**0.783**	**0.846**

Table 4: Ablation experiments on preference sampling adaptation (PSA) and preference-aware hypernet fusion (PHF).

HV on the SYNTHETIC dataset (from 0.719 to 0.793). With PSA, the improvement is smaller (i.e., 0.2%) due to the high global HV had been achieved by PSA (i.e., 0.825). Similar patterns can be observed on the other three datasets.

5 Conclusion

In this paper, we proposed HetPFL, a comprehensive method for learning both local and global Pareto fronts in fair federated learning. First, HetPFL includes a Preference Sampling Adaptation (PSA) approach, which adaptively learns the preference sampling distribution for each client. Second, HetPFL incorporates a Preference-aware Hypernet Fusion (PHF) approach, which guides the generation of the global hypernet by learning the mapping from preferences to fusion weights at the server. We theoretically prove that HetPFL achieves an error convergence rate of order $\mathcal{O}(\frac{1}{t})$. Experimental results demonstrate that HetPFL achieves superior performance in learning both local and global Pareto fronts compared to seven state-of-the-art methods across four datasets.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62202214 and Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515012819.

References

[Barenstein, 2019] Matias Barenstein. Propublica's compas data revisited. *arXiv preprint arXiv:1906.04711*, 2019.

[Collins *et al.*, 2021] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021.

[Deng *et al.*, 2020] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Distributionally robust federated averaging. In *Proceedings of Advances in Neural Information Processing Systems*, volume 33, pages 15111–15122, 2020.

[Du *et al.*, 2021] Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 181–189. SIAM, 2021.

[Dua *et al.*, 2017] Dheeru Dua, Casey Graff, et al. Uci machine learning repository, 2017. *URL http://archive. ics. uci. edu/ml*, 7(1):62, 2017.

[Ezzeldin *et al.*, 2023] Yahya H Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and A Salman Avestimehr. Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7494–7502, 2023.

[Feldman *et al.*, 2015] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

[Hong *et al.*, 2023] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023.

[Imteaj and Amini, 2022] Ahmed Imteaj and M Hadi Amini. Leveraging asynchronous federated learning to predict customers financial distress. *Intelligent Systems with Applications*, 14:200064, 2022.

[Kamishima *et al.*, 2012] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*, pages 35–50. Springer, 2012.

[Li *et al.*, 2019] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[Lin *et al.*, 2022] Xi Lin, Zhiyuan Yang, and Qingfu Zhang. Pareto set learning for neural multi-objective combinatorial optimization. In *Proceedings of International Conference on Learning Representations*, 2022.

[Lyu *et al.*, 2020] Lingjuan Lyu, Xinyi Xu, Qian Wang, and Han Yu. Collaborative fairness in federated learning. *Federated Learning: Privacy and Incentive*, pages 189–204, 2020.

[Makhija *et al.*, 2024] Disha Makhija, Xing Han, Joydeep Ghosh, and Yejin Kim. Achieving fairness across local and global models in federated learning. *arXiv preprint arXiv:2406.17102*, 2024.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[Miettinen, 1999] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 1999.

[Moro *et al.*, 2014] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

[Nguyen *et al.*, 2021] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, and H Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658, 2021.

[Rieke *et al.*, 2020] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

[Roh *et al.*, 2020] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. Fairbatch: Batch selection for model fairness. *arXiv preprint arXiv:2012.01696*, 2020.

[Salazar *et al.*, 2023] Teresa Salazar, Miguel Fernandes, Helder Araújo, and Pedro Henriques Abreu. Fair-fate: Fair federated learning with momentum. In *Proceedings of International Conference on Computational Science*, pages 524–538. Springer, 2023.

[Salimans *et al.*, 2017] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[Thapa *et al.*, 2022] Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8485–8493, 2022.

[Wang *et al.*, 2021] Zheng Wang, Xiaoliang Fan, Jianzhong Qi, Chenglu Wen, Cheng Wang, and Rongshan Yu. Federated learning with fair averaging. *arXiv preprint arXiv:2104.14937*, 2021.

[Ye *et al.*, 2025] Rongguang Ye, Wei-Bin Kou, and Ming Tang. Praffl: A preference-aware scheme in fair federated learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, page 1797–1808, 2025.

[Yue *et al.*, 2023] Xubo Yue, Maher Nouiehed, and Raed Al Kontar. Gifair-fl: A framework for group and individual fairness in federated learning. *INFORMS Journal on Data Science*, 2(1):10–23, 2023.

[Zeng *et al.*, 2021] Yuchen Zeng, Hongxu Chen, and Kangwook Lee. Improving fairness via federated learning. *arXiv preprint arXiv:2110.15545*, 2021.

[Zeng *et al.*, 2023] Yuchen Zeng, Hongxu Chen, and Kangwook Lee. Federated learning with local fairness constraints. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 1937–1942. IEEE, 2023.

[Zitzler and Thiele, 1999] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.