Few-for-Many Personalized Federated Learning

Ping Guo[1,6], Tiantian Zhang[2*], Xi Lin[3], Xiang Li[4], Zhi-Ri Tang[5], Qingfu Zhang[1,6†]
[1]City University of Hong Kong; [2]Hong Kong Metropolitan University;
[3]Xi'an Jiaotong University; [4]Southeast University; [5]Jinan University;
[6]CityU Shenzhen Research Institute

Abstract

Personalized Federated Learning (PFL) aims to train customized models for clients with highly heterogeneous data distributions while preserving data privacy. Existing approaches often rely on heuristics like clustering or model interpolation, which lack principled mechanisms for balancing heterogeneous client objectives. Serving M clients with distinct data distributions is inherently a multi-objective optimization problem, where achieving optimal personalization ideally requires M distinct models on the Pareto front. However, maintaining M separate models poses significant scalability challenges in federated settings with hundreds or thousands of clients. To address this challenge, we reformulate PFL as a few-for-many optimization problem that maintains only K shared server models ($K \ll M$) to collectively serve all M clients. We prove that this framework achieves near-optimal personalization: the approximation error diminishes as K increases and each client's model converges to each client's optimum as data grows. Building on this reformulation, we propose FedFew, a practical algorithm that jointly optimizes the K server models through efficient gradient-based updates. Unlike clustering-based approaches that require manual client partitioning or interpolation-based methods that demand careful hyperparameter tuning, FedFew automatically discovers the optimal model diversity through its optimization process. Experiments across vision, NLP, and real-world medical imaging datasets demonstrate that FedFew, with just 3 models, consistently outperforms other state-of-the-art approaches. Code is available at https://github.com/pgg3/FedFew.

1. Introduction

Personalized Federated Learning (PFL) [37] aims to train client-specific models that best fit each client's local data distribution by leveraging aggregated knowledge from collaborative learning across the federation. This paradigm overcomes a fundamental limitation of traditional Federated Learning (FL) [29], where a single global model struggles to effectively serve all clients when their data are drawn from vastly different distributions P_i under the non-IID setting. The benefit of personalized collaborative learning has made PFL particularly valuable in domains such as healthcare [28] and finance [3], where heterogeneous data distributions necessitate client-specific models while privacy constraints require decentralized training.

The heterogeneity of client data changes the nature of the optimization landscape in PFL [19, 39]. When clients have distinct data distributions $P_i \neq P_j$, a model update that benefits one client may harm another due to their inherent conflicting objectives [45]. For instance, in a healthcare federation where hospitals in urban and rural areas have vastly different patient demographics, optimizing model accuracy for urban hospitals might learn feature representations that poorly capture rural patient characteristics [11, 24].

This inherent conflict naturally leads to a multi-objective optimization perspective [14, 37]. Rather than seeking a single consensus model, PFL must navigate trade-offs among M distinct client loss functions $\{L_1, L_2, \ldots, L_M\}$, where each client i requires a model tailored to its local data distribution P_i. While achieving optimal personalization would ideally require learning M distinct models, maintaining and training such a large number of separate models introduces prohibitive scalability challenges in federated settings involving hundreds or thousands of clients.

Given this scalability challenge, existing PFL methods struggle to effectively balance personalization and computational efficiency. Methods that explicitly adopt the multi-objective perspective, such as FedMGDA [14] and FedMTL [37], only obtain a single model on the *Pareto front* [30], which is the set of all optimal trade-off solutions. Consequently, they cannot provide individual optima for each client. Meanwhile, most PFL methods resort to heuristics without theoretical Pareto optimality guarantees: clustering-based methods like IFCA [10] and CFL [35]

*Corresponding author: ttzhang@hkmu.edu.hk
†Corresponding author: qingfu.zhang@cityu.edu.hk

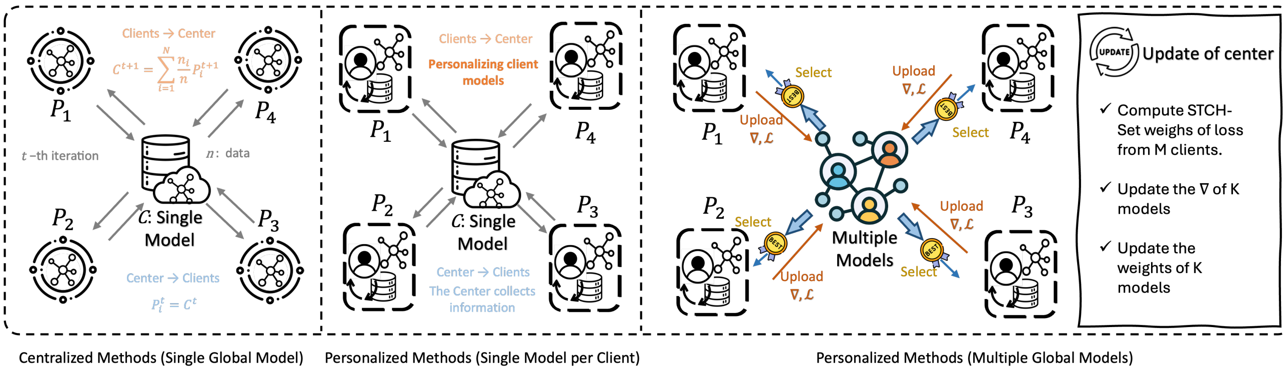


Figure 1. **Paradigms of Personalized Federated Learning. Left:** Centralized methods maintain a single global model for all M clients, failing to capture client heterogeneity. **Center:** Per-client methods train M independent models, sacrificing collaborative learning benefits and suffering from data scarcity. **Right:** Our proposed few-for-many approach maintains K server models ($K \ll M$) that collectively serve all clients. Each client selects the best-fitting model, achieving strong personalization while preserving collaboration.

train one model per group; interpolation-based methods like APFL [6] and Ditto [20] mix global and local models using ad-hoc weights; and per-client methods like FedRep [4] train M independent models, thereby sacrificing collaboration benefits. In summary, existing approaches face a significant limitation: *multi-objective methods produce only a single Pareto-optimal model without personalization, while heuristic methods generate M personalized models without Pareto optimality guarantee.*

To address this challenge, we reformulate PFL as a few-for-many optimization problem that maintains only K server models that collectively serve all M clients (where $K \ll M$) as illustrated in Figure 1, where each client selects the model that best fits its local data distribution. We rigorously prove that the K-for-M framework achieves near-optimal personalization through a precise error decomposition. Our analysis establishes two vanishing error components: the *Pareto coverage gap* from using $K < M$ models diminishes as K increases, and the *statistical error* between empirical and population losses vanishes as client dataset sizes grow. These dual convergence guarantees distinguish our approach from existing heuristic PFL methods.

Building upon this theoretical foundation, we propose **FedFew** (Federated Learning with Few Models), a novel algorithm that enables efficient gradient-based optimization in federated settings. The core challenge lies in jointly optimizing the K server models alongside discrete client-model assignments, which is incompatible with standard gradient-based methods. We address this through a two-level smoothing technique that transforms the discrete selection problem into a fully differentiable objective. This formulation enables clients to perform soft model selection via gradient descent, while the server jointly updates all K models to collaboratively cover all client needs. Unlike clustering-based approaches that require manual client partitioning or interpolation-based methods that demand careful hyperparameter tuning, FedFew can automatically discover the optimal model diversity through its optimization process. The algorithm seamlessly integrates with standard federated learning protocols, incurring only minimal communication overhead compared to single-model training.

Our contributions are three-fold:

- We introduce the few-for-many optimization framework that reformulates PFL as maintaining K shared models ($K \ll M$) to serve M clients, addressing the scalability challenge with rigorous convergence guarantees through Pareto coverage gap and statistical error decomposition.
- We develop FedFew, a practical federated algorithm that solves the few-for-many problem via two-level smoothing, enabling automatic model diversity discovery through gradient-based optimization without manual client clustering or delicate hyperparameter tuning.
- We demonstrate through extensive experiments on seven datasets, including real-world medical imaging application, that FedFew consistently outperforms existing methods while utilizing only 3 models.

2. Related Work

2.1. Standard and Personalized Federated Learning

Standard Federated Learning. Traditional approaches in federated learning aim to learn a single global model by aggregating local updates from distributed clients. Starting with FedAvg [29], which introduced weighted averaging, subsequent methods like FedProx [19], SCAFFOLD [16], and FedDyn [1] have brought improvements in training convergence and stability.

While FL methods perform well under the IID data assumption, real-world FL problems often exhibit significant data heterogeneity across clients. This mismatch can lead to degraded performance and slow convergence [18, 45]. Therefore, personalized federated learning approaches have been proposed to address this challenge by tailoring models to individual client distributions while still leveraging collaborative learning benefits.

Personalized Federated Learning. Existing PFL methods can be roughly grouped into three categories: centralized methods with a single global model, personalized methods with one model per client, and personalized methods with multiple server models.

Centralized Methods (Single Global Model). While the standard FL methods mentioned above (FedAvg [29], FedProx [19], etc.) maintain a single global model, they serve as important baselines for evaluating personalized approaches. Their limitation in handling heterogeneous data motivates the development of personalized methods.

Personalized Methods (Single Model per Client). These methods maintain a unique personalized model for each client while leveraging knowledge from other clients. Representation-based approaches like FedRep [4] and FedBABU [32] decouple the model into shared and personalized components. Bi-level optimization methods such as Ditto [20], pFedMe [7], and Per-FedAvg [9] formulate personalization as a nested optimization problem. Model interpolation approaches blend global and local models to achieve personalization. For example, FedBN [21] personalizes only batch normalization layers, while APFL [6] learns explicit weights to mix global and local models. More recent methods like FedFomo [42] and FedAMP [15] investigate adaptive mixing strategies.

Personalized Methods (Multiple Server Models). These methods learn a small set of specialized models on the server by grouping clients with similar data distributions. IFCA [10] and CFL [35] employ explicit hard clustering algorithms to assign each client to one model cluster. More recent approaches like FedSoft [34], FeSEM [27], and PACFL [38] leverage soft clustering or expectation-maximization techniques for more flexible client-model associations. However, these methods lack theoretical guarantees on the quality of the learned model set and rely on heuristic clustering objectives.

2.2. Multi-Objective Optimization in FL

Multi-Objective Optimization. Classical multi-objective optimization (MOO) approaches, such as weighted sum, Tchebycheff scalarization, and Normal Boundary Intersection [43], aim to identify a set of Pareto-optimal solutions with various trade-offs among objectives. Recent gradient-based methods like MGDA [8], PCGrad [40], and ParetoMTL [22] address multi-objective optimization by balancing conflicting gradients across objectives during the optimization process. Most recently, set scalarization methods [23, 25, 26] have emerged, which propose to approximate the Pareto front with a small solution set.

MOO in Federated Learning. Several PFL approaches have recognized the multi-objective nature of federated learning and attempted to address it through multi-task learning frameworks. FedMTL [37] leverages task relationship matrices to enable knowledge transfer between clients. More recently, FedMGDA [14] adopts Multiple Gradient Descent Algorithm (MGDA) to balance conflicting client objectives by finding a common gradient direction that benefits all clients. While theoretically principled, these approaches involve complex bi-level optimization procedures and incur significant communication overhead, making them computationally prohibitive for large-scale federated settings. Moreover, these methods target a single trade-off solution, limiting their ability to handle heterogeneous client preferences.

3. PFL as Multi-Objective Optimization

3.1. Problem Setup and Client Objectives

Consider a federated learning system with M clients, where each client i possesses a local dataset D_i drawn from a distinct distribution P_i. The goal of each client is to find a model that minimizes its expected loss:

$$\begin{aligned} \theta_i^* &= \arg\min_{\theta} L_i(\theta), \\ \text{where} \quad L_i(\theta) &= \mathbb{E}_{(x,y)\sim P_i}[\ell(f(x;\theta),y)] \end{aligned} \tag{1}$$

where $\ell : \mathbb{R}^d \times \mathcal{Y} \to \mathbb{R}$ is the loss function and $f(\cdot;\theta)$ denotes the model parameterized by θ.

In practice, clients work with empirical risk minimization over their finite local datasets:

$$\hat{L}_i(\theta) = \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \ell(f(x;\theta),y). \tag{2}$$

This setting reveals two key challenges of the PFL problem: *Collaboration* and *Multi-Objective Trade-offs*.

Collaboration. Independent local training often leads to severe overfitting due to limited data availability at each client, with generalization error scaling as $\mathcal{O}(1/\sqrt{|\mathcal{D}_i|})$ [36]. This limitation motivates collaborative learning that leverages data from other clients.

Multi-Objective Trade-Offs. However, collaboration introduces a challenge: when client data distributions are heterogeneous (i.e., $P_i \neq P_j, \forall i \neq j$), optimizing for one client may degrade performance on others. This inherent conflict reveals that PFL is intrinsically an M-objective optimization problem:

$$\min_{\theta} \mathbf{L}(\theta) = [L_1(\theta), L_2(\theta), \ldots, L_M(\theta)]^T \tag{3}$$

where no single model θ can simultaneously minimize all objectives. To characterize optimal solutions, we introduce the concept of Pareto optimality:

Definition 3.1 (Pareto Optimality [30])**.** A model θ^* is *Pareto optimal* if there exists no other model θ such that

$L_i(\theta) \le L_i(\theta^*)$ for all $i \in [M]$ with strict inequality for at least one client. The *Pareto set* contains all Pareto optimal models, and the *Pareto front* is the set of objective vectors $\{[L_1(\theta^*), \dots, L_M(\theta^*)]^T : \theta^* \text{ is Pareto optimal}\}$.

While the Pareto front contains all optimal trade-off models, directly approximating it with high precision becomes computationally prohibitive. Specifically, achieving ε-accuracy, where every Pareto optimal point has its representative within ε distance, requires $\mathcal{O}((1/\varepsilon)^{M-1})$ models [5]. This exponential dependence on M renders direct approximation infeasible: even with $M = 10$ clients, achieving $\varepsilon = 0.01$ requires 10^{18} models, which is far beyond any practical system's capacity.

A Key Insight. While the Pareto front is continuous and requires exponential models to fully approximate, achieving optimal personalization does not require approximating the entire front. Instead, we only need to find M distinct models on the Pareto front, with one tailored for each client's distribution. However, maintaining M separate models remains impractical: the communication and computation costs grow linearly with M, becoming prohibitive when serving hundreds or thousands of clients.

This motivates our practical K-for-M framework: we maintain only K models (where $K \ll M$) that collectively serve all clients. Each client selects the best-fitting model from this set, achieving effective personalization with tractable overhead.

3.2. Set-based Optimization: K-for-M Framework

K-for-M Reformulation. Let $\Theta = \{\theta_1, \dots, \theta_K\}$ denote the set of K models maintained by the server. Each client i will be served by the model θ_{k_i} that minimizes its local loss $L_i(\theta_{k_i})$. This transforms the original multi-objective problem (3) into[1]:

$$\min_{\Theta} \mathbf{F}(\Theta) = \begin{bmatrix} \min\limits_{k_1 \in \{1,\dots,K\}} L_1(\theta_{k_1}) \\ \vdots \\ \min\limits_{k_M \in \{1,\dots,K\}} L_M(\theta_{k_M}) \end{bmatrix}. \tag{4}$$

The framework provides a natural mechanism for quality control: by adjusting K, system designers can systematically trade off between personalization quality and computational cost.

Impact of K. The choice of K determines the trade-off between personalization capacity and system efficiency:

- $K = 1$: Degenerates to a single model, where global model training (*e.g.,* FedAvg) can find one Pareto optimal solution but fails to provide personalization;
- $K = M$: Each client could potentially have its own personalized model;
- $1 < K < M$: Our operating regime, balancing personalization quality with communication efficiency.

[1]Several existing PFL methods (*e.g.,* IFCA [10]) implicitly tackle this same K-for-M optimization problem, though with different solution approaches.

Convergence Analysis. The following theorem characterizes the convergence rate in terms of two error components: the Pareto coverage gap and the statistical error.

Theorem 3.1 (Convergence of K-for-M Framework)**.** Let $\Theta^{(K)} = \{\theta_1, \dots, \theta_K\}$ be the optimal solution with K models for M clients. Define $\Delta_{het} = \max_{i,j \in [M]}[L_i(\theta_j^*) - L_i(\theta_i^*)]$ as the maximum pairwise heterogeneity. Then the average error across clients is bounded by:

$$\begin{aligned} &\frac{1}{M}\sum_{i=1}^{M}\left\{\mathbb{E}\left[\min_{k\in[K]} L_i(\theta_k)\right] - L_i(\theta_i^*)\right\} \\ &\le \underbrace{\frac{M-K}{M}\cdot\Delta_{het}}_{\text{Pareto coverage gap}} + \underbrace{\mathcal{O}\left(\sqrt{\frac{Kd}{n}}\right)}_{\text{statistical error}} \end{aligned} \tag{5}$$

where $\theta_i^* = \arg\min_\theta L_i(\theta)$ is client i's optimal personalized model, d is the model complexity, and n is the average sample size per client. The complete proof is provided in the supplementary material.

Remark 3.1 (Convergence to Optimal Solution)**.** The bound decomposes the approximation error into two independent dimensions: (i) when $K = M$, the Pareto coverage gap vanishes, recovering individual personalized models for each client; (ii) as the local dataset size $n \to \infty$, the statistical error vanishes, ensuring the empirical solution converges to the population optimum. Achieving zero error requires both conditions simultaneously.

4. FedFew Algorithm

4.1. Smooth Tchebycheff Set Scalarization

The K-for-M formulation (4) is an M-objective optimization problem, where each objective involves selecting the best model from a set Θ. To solve this problem while ensuring Pareto optimality, we adopt the Tchebycheff set scalarization (TCH-Set) approach, which transforms the multi-objective problem into a single scalar objective [23, 43]:

$$g^{\text{TCH-Set}}(\Theta|\lambda) = \max_{1\le i\le M}\left\{\lambda_i\left(\min_{1\le k\le K} L_i(\theta_k) - z_i^*\right)\right\} \tag{6}$$

where $\lambda = (\lambda_1, \dots, \lambda_M)$ are client preference weights and z_i^* is the ideal loss value for client i.

This scalarization is particularly suited for personalized federated settings because: (i) it guarantees Pareto optimality of the solutions and (ii) it naturally handles heterogeneous client objectives without requiring explicit aggregation. However, the nested $\max$ and $\min$ operators make (6) non-differentiable, preventing gradient-based optimization required for federated training.

Two-Level Smoothing. Since both max and min operators are non-differentiable, we employ log-sum-exp smoothing to enable gradient-based optimization [12, 23]:

$$\max_i\{x_i\} \approx \mu \log\left(\sum_i \exp(x_i/\mu)\right) \tag{7}$$

$$\min_i\{x_i\} \approx -\mu \log\left(\sum_i \exp(-x_i/\mu)\right) \tag{8}$$

where $\mu > 0$ controls the approximation quality.

Final Formulation. For simplicity, we set $\lambda_i = 1$ and $z_i^* = 0$. Applying the two-level smoothing (7) and (8) to (6), we obtain the smooth Tchebycheff set scalarization (STCH-Set):

$$g^{\text{STCH-Set}}(\Theta) = \mu \log \sum_{i=1}^{M} \left(\sum_{k=1}^{K} \exp\left(-\frac{L_i(\theta_k)}{\mu}\right)\right)^{-1} \tag{9}$$

where $\mu > 0$ is the smoothing parameter.[2]

4.2. Decomposed Gradient Computation

Taking the gradient of $g^{\text{STCH-Set}}$ in (9) with respect to θ_k:

$$\nabla_{\theta_k} g^{\text{STCH-Set}} = \sum_{i=1}^{M} \alpha_i \cdot w_{ik} \cdot \nabla_{\theta_k} L_i(\theta_k) \tag{10}$$

where the weights decompose into two components. Define $S_i = \sum_{k=1}^{K} \exp(-L_i(\theta_k)/\mu)$. Then:

$$\text{(Outer weight)} \quad \alpha_i = \frac{S_i^{-1}}{\sum_{j=1}^{M} S_j^{-1}}, \tag{11}$$

$$\text{(Inner weight)} \quad w_{ik} = \frac{\exp(-L_i(\theta_k)/\mu)}{S_i}. \tag{12}$$

The outer weight α_i assigns higher importance to clients with larger S_i^{-1} (i.e., clients that perform poorly across all models), thereby implementing a hard-sample mining effect. The inner weight w_{ik} performs soft model selection by assigning higher weights to models with lower loss for each client i, thus smoothly identifying the best-matching model for that client.

4.3. Federated Implementation

FedFew alternates between client gradient computation and server model updates through smooth Tchebycheff set scalarization. The federated optimization proceeds in communication rounds as outlined in Algorithm 1.

[2] In implementation, we weight each client's loss $L_i(\theta_k)$ by its normalized sample size to account for varying local dataset sizes like FedAvg [29] before aggregation, i.e., $L_i(\theta_k) \leftarrow \frac{n_i}{\sum_{j=1}^{M} n_j} \cdot L_i(\theta_k)$.

Algorithm 1 FedFew: Few-for-Many PFL with Smooth Tchebycheff Set Scalarization

Input: M clients with datasets $\{\mathcal{D}_i\}_{i=1}^M$, K initial models $\Theta^{(0)}$, smoothing parameter μ, learning rate η, local epochs E, communication rounds T
Output: Optimized model set $\Theta^{(T)}$
for $t = 1, 2, \dots, T$ **do**
  Server broadcasts $\Theta^{(t-1)}$ to all clients
  for each client $i = 1, 2, \dots, M$ **in parallel do**
    for each model $k = 1, 2, \dots, K$ **do**
      for $e = 1, 2, \dots, E$ **do**
        Update local model: $\theta_k \leftarrow \theta_k - \eta \nabla_{\theta_k} L_i(\theta_k)$
      end for
      Compute gradient $g_{ik}^{(t)} = \nabla_{\theta_k} L_i(\theta_k)$ and loss $L_i^{(t)}(\theta_k)$
    end for
    Send $\{(g_{ik}^{(t)}, L_i^{(t)}(\theta_k))\}_{k=1}^K$ to server
  end for
  for each model $k = 1, 2, \dots, K$ **do**
    Compute STCH-Set weights $\{\alpha_i, w_{ik}\}$ from losses $\{L_i^{(t)}(\theta_k)\}$ using Eqs. (11) and (12)
    $\nabla_{\theta_k} g^{\text{STCH-Set}} = \sum_{i=1}^M \alpha_i \cdot w_{ik} \cdot g_{ik}^{(t)}$
    $\theta_k^{(t)} = \theta_k^{(t-1)} - \eta \cdot \nabla_{\theta_k} g^{\text{STCH-Set}}$
  end for
end for
Model Selection (post-training):
Server broadcasts trained models $\Theta^{(T)}$ to all clients
for each client $i = 1, 2, \dots, M$ **in parallel do**
  Client i evaluates all K models on local validation/training data
  Compute losses: $\{L_i(\theta_1^{(T)}), L_i(\theta_2^{(T)}), \dots, L_i(\theta_K^{(T)})\}$
  Select best model: $k_i^* = \arg\min_{k \in \{1,\dots,K\}} L_i(\theta_k^{(T)})$
end for

Client Side: Each client i computes local gradients $g_{ik} = \nabla_{\theta_k} L_i(\theta_k)$ for all K models and sends them to the server.

Server Side: The server computes weights $\{\alpha_i, w_{ik}\}$ from current losses $\{L_i(\theta_k)\}$ and aggregates:

$$\nabla_{\theta_k} g^{\text{STCH-Set}} = \sum_{i=1}^{M} \alpha_i \cdot w_{ik} \cdot g_{ik}. \tag{13}$$

Model Selection Mechanism. After training, each client identifies the most suitable model from the K available candidates through a simple local evaluation procedure. This process involves performing forward passes with all K models on the client's local validation or training set, computing the corresponding losses, and selecting the model that achieves the minimum loss.

Communication Efficiency. In each communication round, every client performs E local epochs of training and sends K gradients along with K scalar loss values to the server. The per-client communication cost is $\mathcal{O}(Kd)$ where d is the model dimension. Since K is typically small (ranging from 3 to 10 in practice) and remains fixed regardless of the total number of clients M, the resulting communication overhead factor of K is modest. More importantly, by using local epochs $E > 1$, the number of required communication rounds T can be reduced proportionally (See § 5.3.2).

4.4. Convergence Guarantees

We establish two key theoretical properties: uniform approximation quality and Pareto optimality guarantees.

Theorem 4.1 (Uniform Smooth Approximation [23])**.** The smooth Tchebycheff set scalarization $g^{\text{STCH-Set}}(\Theta)$ uniformly approximates the non-smooth version $g^{\text{TCH-Set}}(\Theta)$. As the smoothing parameter $\mu \to 0$:

$$\lim_{\mu \to 0} g^{\text{STCH-Set}}(\Theta) = g^{\text{TCH-Set}}(\Theta) \tag{14}$$

uniformly over all model sets Θ, with approximation error bounded by $\mathcal{O}(\mu \log M + \mu \log K)$.

The smoothing parameter μ controls the degree of smoothness in the approximation: smaller μ yields a tighter approximation to the original min-max objective, but results in sharper gradients that may hinder optimization.

Theorem 4.2 (Pareto Properties of STCH-Set [23])**.** The smooth Tchebycheff set scalarization provides strong Pareto guarantees:

1. **Pareto Optimality:** All solutions in the optimal set Θ^*_K are weakly Pareto optimal. Moreover, they are Pareto optimal if either the optimal set is unique or all preference coefficients are positive.
2. **Pareto Stationarity:** If gradient descent converges to $\hat{\Theta} = \{\hat{\theta}_1, \ldots, \hat{\theta}_K\}$ where $\nabla_{\hat{\theta}_k} g^{\text{STCH-Set}}(\hat{\Theta}) = 0$ for all k, then all solutions in $\hat{\Theta}$ are Pareto stationary for the original multi-objective problem (3).

Combined with standard SGD convergence analysis, gradient descent on the smooth objective drives the expected squared gradient norm to $\mathcal{O}(1/\sqrt{T})$ after T iterations. The overall approximation quality is controlled by both the optimization error ($\sim 1/\sqrt{T}$) and the smoothing error ($\sim \mu \log M + \mu \log K$). Detailed proofs are provided in the supplementary material.

Comparison with Clustering. Interestingly, clustering-based methods like IFCA [10] attempt to solve the same K-for-M optimization problem (4). However, its hard client-to-cluster assignment creates a non-convex, discontinuous optimization landscape that lacks convergence guarantees. In contrast, our smooth Tchebycheff formulation ensures convergence to Pareto stationary points (Theorem 4.2), demonstrating that the optimization strategy is crucial for both theoretical guarantees and practical performance.

5. Experiments

5.1. Experimental Setup

Datasets. We evaluate our method on benchmark datasets with controlled heterogeneity and real-world medical imaging datasets under two distinct settings: *pathological* (extreme label imbalance) and *practical* (realistic label skew via Dirichlet distribution or natural partitions).

In benchmark datasets, under the *pathological setting*, we use CIFAR-100 [17] partitioned by assigning 2 classes per client ($M \in \{10, 20\}$), creating extreme label imbalance. Under the *practical setting*, we partition data using Dirichlet ($\alpha = 0.5$) distribution [45], where smaller α induces stronger label skew. Specifically, we evaluate on CIFAR-10 [17] and CIFAR-100 ($M \in \{10, 20\}$ clients), TinyImageNet ($M = 10$), and AG News [44] ($M = 20$). We also include FEMNIST [2] ($M = 20$) with natural user-based partitioning.

We further validate our method on real-world medical datasets, where data heterogeneity arises from natural domain shifts across medical institutions. Kvasir [33] is a gastrointestinal disease detection dataset containing endoscopic images across 8 classes (polyps, ulcerative colitis, etc.), which we partition among $M = 5$ clients using Dirichlet ($\alpha = 0.5$) to simulate hospitals with different disease prevalence. FedISIC2019 [31] is a skin lesion classification dataset from the ISIC 2019 challenge with 8 diagnostic categories, where the data naturally originates from $M = 6$ different medical centers, each with distinct imaging equipment and patient demographics, creating realistic cross-institutional heterogeneity.

Baselines. We compare FedFew against nine baseline approaches spanning different personalization strategies. For *centralized methods*, we include FedAvg [29] and FedProx [19], which train a single global model shared by all clients. We also include FedMTL [37], a multi-task learning approach that learns task relationships to enable model personalization. For *personalized methods with single model per client*, we evaluate APFL [6], Ditto [20], FedRep [4], and FedAMP [15], which maintain separate personalized models for each client. For *personalized methods with multiple server models*, we compare against IFCA [10], our core competitor that also uses K shared models but relies on hard clustering. For medical datasets, we further include a *local-only baseline* trained solely on local data without federation to assess the benefit of collaborative learning.

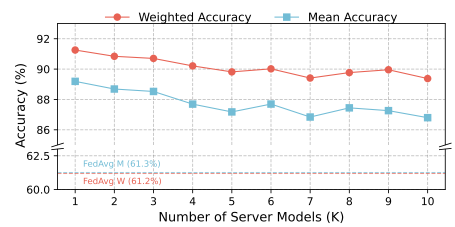
Implementation. We employ a 4-layer CNN [29] for CIFAR-10/100, FEMNIST, and TinyImageNet, TextCNN [41] for AG News, and ResNet-18 [13] for

Table 1. Overall performance comparison across CIFAR-10, CIFAR-100, TINY (TinyImageNet), AG News, and FEMNIST datasets under pathological and practical heterogeneous settings. Results are reported as mean accuracy (%) $\pm$ standard deviation. Best results are **highlighted** and **bolded**, second-best results are underlined.

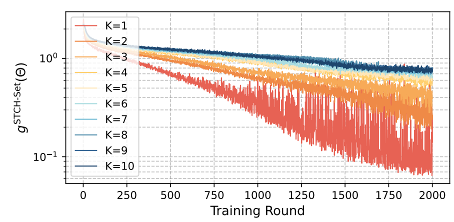
Method	Pathological heterogeneous setting		Practical heterogeneous setting						
	CIFAR-100		FEMNIST	CIFAR-10		CIFAR-100		TINY	AG News
	$M=10$	$M=20$	$M=20$	$M=10$	$M=20$	$M=10$	$M=20$	$M=10$	$M=20$
Centralized Methods (Single Global Model)									
FedAvg [29]	29.00 ± 3.94	28.57 ± 4.37	96.65 ± 1.81	61.36 ± 8.54	61.26 ± 8.64	30.84 ± 2.26	31.10 ± 4.21	13.49 ± 1.55	88.90 ± 7.15
FedProx [19]	28.56 ± 4.50	28.11 ± 3.86	96.51 ± 2.32	60.87 ± 7.87	61.38 ± 9.35	30.74 ± 2.25	30.90 ± 4.03	13.64 ± 1.56	83.42 ± 11.34
FedMTL [37]	65.33 ± 3.64	59.65 ± 3.70	**100.00 ± 0.00**	85.92 ± 11.18	85.75 ± 11.62	46.28 ± 3.82	44.79 ± 5.25	23.49 ± 2.43	94.10 ± 7.56
Personalized Methods (Single Model per Client)									
APFL [6]	64.69 ± 4.01	59.97 ± 3.91	99.93 ± 0.20	**88.38 ± 7.83**	87.36 ± 10.92	48.30 ± 3.44	46.67 ± 5.11	24.26 ± 2.71	94.26 ± 7.41
Ditto [20]	65.32 ± 3.63	59.61 ± 3.66	**100.00 ± 0.00**	85.97 ± 10.95	85.72 ± 11.77	46.19 ± 3.50	44.89 ± 5.16	23.45 ± 2.70	94.06 ± 7.69
FedRep [4]	**66.50 ± 3.41**	61.46 ± 3.82	**100.00 ± 0.00**	87.36 ± 8.92	86.94 ± 10.76	48.26 ± 3.31	46.46 ± 4.52	27.24 ± 2.77	94.68 ± 12.69
FedAMP [15]	65.41 ± 3.67	59.66 ± 3.59	**100.00 ± 0.00**	85.96 ± 10.92	85.93 ± 11.53	46.13 ± 3.62	45.00 ± 5.08	23.32 ± 2.62	94.17 ± 7.48
Personalized Methods (Multiple Server Models)									
IFCA [10]	42.34 ± 5.18	43.89 ± 3.58	99.46 ± 0.78	77.27 ± 7.14	73.35 ± 12.00	34.09 ± 5.69	29.80 ± 3.75	15.24 ± 1.90	90.63 ± 11.79
FedFew (Ours)	65.47 ± 3.90	**64.98 ± 3.32**	**100.00 ± 0.00**	88.17 ± 7.74	**88.26 ± 9.06**	**50.44 ± 3.14**	**53.69 ± 4.79**	**30.31 ± 3.06**	**96.07 ± 4.82**

Table 2. Performance comparison on medical imaging datasets (Kvasir and FedISIC). For each dataset, we report three metrics: (1) Avg: average accuracy across all clients with standard deviation; (2) Min: worst-case client accuracy; (3) Max: best-case client accuracy. Best results are **highlighted** and **bolded**, second-best results are underlined.

Method	Kvasir			FedISIC		
	Avg. ± Std.	*Min.*	*Max.*	*Avg. ± Std.*	*Min.*	*Max.*
Local-only Baseline						
Local-only	92.16 ± 7.22	80.49	**100.00**	65.37 ± 17.23	41.08	94.54
Centralized Methods (Single Global Model)						
FedAvg [29]	85.96 ± 2.30	82.20	89.20	64.71 ± 15.66	48.50	95.15
FedProx [19]	79.91 ± 11.24	57.51	86.58	65.46 ± 18.77	42.93	96.06
FedMTL [37]	92.46 ± 6.75	82.20	**100.00**	69.20 ± 15.29	54.19	96.46
Personalized Methods (Single Model per Client)						
APFL [6]	91.97 ± 7.00	82.20	99.77	67.83 ± 15.63	52.92	95.45
Ditto [20]	92.37 ± 7.17	80.73	**100.00**	69.51 ± 15.72	52.20	96.66
FedRep [4]	92.71 ± 6.47	82.93	**100.00**	64.38 ± 16.51	47.96	94.54
FedAMP [15]	92.76 ± 6.72	82.20	**100.00**	67.41 ± 17.12	45.84	**96.76**
Personalized Methods (Multiple Server Models)						
IFCA [10]	82.05 ± 21.88	40.24	**100.00**	53.61 ± 20.45	23.23	85.74
FedFew (Ours)	**92.84 ± 6.08**	**83.90**	99.77	**69.57 ± 14.59**	**55.40**	95.35



(a) Impact of K



(b) Convergence of $g^{\text{STCH-Set}}$

Figure 2. **Sensitivity Studies.** (a) Test accuracy vs K on CIFAR-10. FedAvg baselines (dashed) shown for comparison. (b) Evolution over training rounds (log scale) for different K values.

medical datasets. Our method utilizes $K = 3$ server models across all experiments. Training proceeds for 2000 communication rounds on benchmark datasets and 1000 rounds on medical datasets to mitigate overfitting on smaller-scale medical data, with 1 local epoch per round. Batch sizes are configured based on dataset characteristics: 100 for datasets with lower overfitting tendency and 50 for those more susceptible to overfitting. Learning rates are selected according to dataset complexity, ranging from 0.0005 to 0.005. Full client participation is enforced in each communication round. Comprehensive hyperparameter configurations are in the supplementary material.

5.2. Main Results

Benchmark Datasets. Table 1 presents the performance comparison on benchmark datasets under both pathological and practical heterogeneity settings.

Our FedFew method demonstrates superior performance across diverse data distributions and client configurations. In pathological heterogeneous settings with CIFAR-100, FedFew achieves 64.98% accuracy with $M = 20$ clients, outperforming the best personalized baseline (FedRep at 61.46%). Under practical heterogeneous settings, FedFew consistently ranks first or second across all datasets. Notably, on CIFAR-100 with $M = 20$ clients, FedFew achieves 53.69% accuracy, surpassing the best baseline by 7.02%. On TinyImageNet, FedFew improves over the strongest baseline (FedRep) by 3.07%, demonstrating its effectiveness on large-scale image classification. For AG News text classification, FedFew achieves 96.07% accuracy, outperforming FedRep by 1.39%.

Real-world Medical Dataset. Table 2 presents results on medical imaging datasets with naturally heterogeneous distributions. On the Kvasir gastrointestinal dataset, FedFew

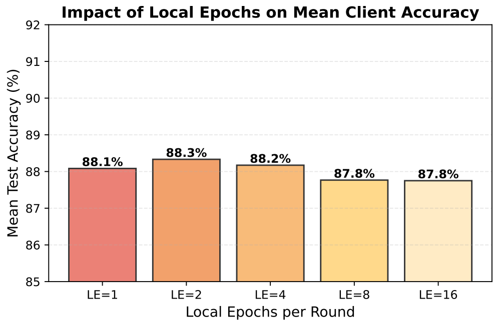


Figure 3. **Mean client accuracy comparison across communication configurations.** All configurations achieve comparable mean client accuracy (87.8–88.3%), demonstrating that our method is robust to different communication-computation trade-offs.

achieves the highest average accuracy (92.84%) and best worst-case performance, demonstrating robustness across diverse medical institutions. For FedISIC skin lesion classification, FedFew attains 69.57% average accuracy with 55.40% minimum accuracy, significantly outperforming IFCA which suffers from severe performance degradation.

Notably, methods adopting multi-objective perspectives (FedFew and FedMTL) both achieve significantly higher minimum accuracies compared to other baselines (at least +1.2% improvement over other baselines, +13.0% over local-only on FedISIC), showcasing the advantage of multi-objective optimization in balancing performance across heterogeneous clients.

5.3. Sensitivity Analysis and Convergence

We conduct sensitivity analysis and convergence studies on CIFAR-10 with Dirichlet-$\alpha = 0.5$ heterogeneity across $M = 20$ clients. Throughout this section, we use $K = 3$ server models by default, except when explicitly varying K to study its impact on performance.

5.3.1. Effect of Number of Server Models

Robust Test Accuracy. Figure 2a presents test accuracy for $K \in \{1, 2, \dots, 10\}$. Our method achieves weighted accuracy ranging from 89.4 to 91.3% across all K values, consistently demonstrating substantial improvement over FedAvg's 61.2% baseline. Notably, the single-model configuration ($K = 1$) attains the highest accuracy of 91.3%. This non-monotonic relationship between K and performance can be attributed to two factors: (1) *Underlying data homogeneity:* CIFAR-10 is sampled from a single distribution, a single well-optimized model can perform well across clients; (2) *Optimization complexity:* larger K expands the parameter space, leading to slower convergence within the fixed training rounds.

Convergence of STCH-Set Objective. To validate this optimization complexity hypothesis, we examine convergence behavior in Figure 2b, which tracks the evolution of $g^{\text{STCH-Set}}(\Theta)$ over 2,000 training rounds (log scale). We observe consistent monotonic decrease across all K values, confirming the stability of our gradient-based optimization. However, with increasing K, the convergence speed decreases significantly, corroborating that larger model sets indeed create more challenging optimization landscapes that hinder both convergence rate and final performance.

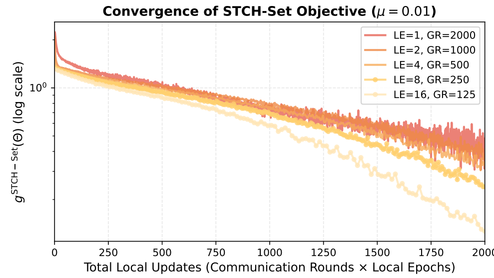


Figure 4. **Communication-computation trade-off.** Convergence of $g^{\text{STCH-Set}}$ vs total local updates for different (local epochs, communication rounds) configurations. Local epochs (LE) $\in \{1, 2, 4, 8, 16\}$ with corresponding communication rounds (GR) to maintain 2000 total updates.

5.3.2. Communication Efficiency

We examine the trade-off between communication frequency and local computation by varying the number of local epochs per round while maintaining constant total local updates (communication rounds × local epochs = 2000).

Figure 4 shows the convergence of $g^{\text{STCH-Set}}(\Theta)$ across five configurations. Configurations with more local epochs (LE=16) exhibit faster convergence and lower variance compared to frequent communication (LE=1). Specifically, LE=16 achieves the steepest descent and most stable optimization trajectory, demonstrating that our method maintains or even improves performance while drastically reducing communication overhead. All configurations reach comparable mean client accuracies as shown in Figure 3.

6. Conclusion

In this paper, we propose FedFew, a novel personalized federated learning algorithm that tackles the scalability challenge in PFL through a Few-for-Many framework, where a small set of K server models collaboratively serve M clients with $K \ll M$. Our approach reformulates PFL as a multi-objective optimization problem and leverages the smooth Tchebycheff set scalarization for effective gradient optimization. Extensive experiments on multiple benchmarks, including healthcare collaborations and edge computing scenarios, demonstrate that FedFew achieves superior personalization performance while maintaining computational efficiency and scalability.

Acknowledgement

The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China [GRF Project No. CityU11215622], and the National Natural Science Foundation of China [Grant No. 62276223, 62401226].

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2

[2] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. *CoRR*, abs/1812.01097, 2018. 6

[3] Pushpita Chatterjee, Debashis Das, and Danda B. Rawat. Federated learning empowered recommendation model for financial consumer services. *IEEE Transactions on Consumer Electronics*, 70(1):2508–2516, 2024. 1

[4] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021. 2, 3, 6, 7

[5] Indraneel Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM J. on Optimization*, 8(3):631–657, 1998. 4

[6] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *CoRR*, abs/2003.13461, 2020. 2, 3, 6, 7

[7] Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 3

[8] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5):313–318, 2012. 3

[9] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: a model-agnostic meta-learning approach. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 3

[10] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 1, 3, 4, 6, 7

[11] Pengfei Guo, Puyang Wang, Jinyuan Zhou, Shanshan Jiang, and Vishal M. Patel. Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2423–2432, 2021. 1

[12] Ping Guo, Cheng Gong, Xi Lin, Fei Liu, Zhichao Lu, Qingfu Zhang, and Zhenkun Wang. MOS-Attack: A Scalable Multi-Objective Adversarial Attack Framework . In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5041–5051, Los Alamitos, CA, USA, 2025. IEEE Computer Society. 5

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[14] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu. Federated learning meets multi-objective optimization. *IEEE Transactions on Network Science and Engineering*, 9 (4):2039–2051, 2022. 1, 3

[15] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 7865–7873. AAAI Press, 2021. 3, 6, 7

[16] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 2

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[18] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection . *IEEE Transactions on Knowledge & Data Engineering*, 35(04):3347–3366, 2023. 2

[19] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, pages 429–450, 2020. 1, 2, 3, 6, 7

[20] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021. 2, 3, 6, 7

[21] Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *International Conference on Learning Representations*, 2021. 3

[22] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto multi-task learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 12060–12070, 2019. 3

[23] Xi Lin, Yilu Liu, Xiaoyuan Zhang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Few for many: Tchebycheff set scalariza-

tion for many-objective optimization. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 3, 4, 5, 6

[24] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. FedDG: Federated Domain Generalization on Medical Image Segmentation via Episodic Learning in Continuous Frequency Space . In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1013–1023, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 1

[25] Yilu Liu, Chengyu Lu, Xi Lin, and Qingfu Zhang. Many-objective cover problem: Discovering few solutions to cover many objectives. In *International Conference on Parallel Problem Solving from Nature*, pages 68–82. Springer, 2024. 3

[26] Yilu Liu, Xi Lin, Liang Zhao, and Qingfu Zhang. Few for many: Towards efficient and flexible many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2025. 3

[27] Guodong Long, Ming Xie, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1):481–500, 2022. 3

[28] Wang Lu, Jindong Wang, Yiqiang Chen, Xin Qin, Renjun Xu, Dimitrios Dimitriadis, and Tao Qin. Personalized federated learning with adaptive batchnorm for healthcare. *IEEE Transactions on Big Data*, 10(6):915–925, 2024. 1

[29] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 3, 5, 6, 7

[30] Kaisa Miettinen. *Nonlinear multiobjective optimization*. Kluwer, 1998. 1, 3

[31] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, Boris Muzellec, Constantin Philippenko, Santiago Silva, Maria Teleńczuk, Shadi Albarqouni, Salman Avestimehr, Aurélien Bellet, Aymeric Dieuleveut, Martin Jaggi, Sai Praneeth Karimireddy, Marco Lorenzi, Giovanni Neglia, Marc Tommasi, and Mathieu Andreux. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. In *Advances in Neural Information Processing Systems*, pages 5315–5334. Curran Associates, Inc., 2022. 6

[32] Jaehoon Oh, SangMook Kim, and Se-Young Yun. FedBABU: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2022. 3

[33] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pal Halvorsen. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection, 2017. 6

[34] Yichen Ruan and Carlee Joe-Wong. Fedsoft: Soft clustered federated learning with proximal local updating. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8124–8131, 2022. 3

[35] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Trans. Neural Networks Learn. Syst.*, 32(8):3710–3722, 2021. 1, 3

[36] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014. 3

[37] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 4427–4437, Red Hook, NY, USA, 2017. Curran Associates Inc. 1, 3, 6, 7

[38] Saeed Vahidian, Mahdi Morafah, Weijia Wang, Vyacheslav Kungurtsev, Chen Chen, Mubarak Shah, and Bill Lin. Efficient distribution similarity identification in clustered federated learning via principal angles between client data subspaces. In *Proceedings of the AAAI conference on artificial intelligence*, pages 10043–10052, 2023. 3

[39] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 1

[40] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 3

[41] Jianqing Zhang, Yang Liu, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Jian Cao. Pfllib: A beginner-friendly and comprehensive personalized federated learning library and benchmark. *Journal of Machine Learning Research*, 26(50):1–10, 2025. 6

[42] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and José M. Álvarez. Personalized federated learning with first order model optimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3

[43] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007. 3, 4

[44] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, page 649–657, Cambridge, MA, USA, 2015. MIT Press. 6

[45] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *CoRR*, abs/1806.00582, 2018. 1, 2, 6