PFedCS: A Personalized Federated Learning Method for Enhancing Collaboration among Similar Classifiers

Siyuan Wu[1], Yongzhe Jia[1], Bowen Liu[1], Haolong Xiang[2*], Xiaolong Xu[2], Wanchun Dou[1*]

[1]State Key Laboratory for Novel Software Technology,
School of Computer Science, Nanjing University, China
[2]School of Software,
Nanjing University of Information Science and Technology, China
{sywu, jiayz, liubw}@smail.nju.edu.cn, {hlxiang, xlxu}@nuist.edu.cn, douwc@nju.edu.cn

Abstract

Personalized federated learning (PFL) has recently gained significant attention for its capability to address the poor convergence performance on highly heterogeneous data and the lack of personalized solutions of traditional federated learning (FL). Existing mainstream approaches either perform personalized aggregation based on a specific model architecture to leverage global knowledge or achieve personalization by exploiting client similarities. However, the former overlooks the discrepancies in client data distributions by indiscriminately aggregating all clients, while the latter lacks fine-grained collaboration of classifiers relevant to local tasks. In view of this challenge, we propose a Personalized Federated learning method for Enhancing Collaboration among Similar Classifiers (**PFedCS**), which aims at improving the client's accuracy on local tasks. Concretely, it is achieved by leveraging awareness of the client classifier similarities to address the above problems. By iteratively measuring the distance of the classifier parameters between clients and clustering with each client as a cluster center, the central server adaptively identifies the collaborating clients with similar data distributions. In addition, a distance-constrained aggregation method is designed to generate customized collaborative classifiers to guide local training. As a result, extensive experimental evaluations conducted on various datasets demonstrate that our method achieves state-of-the-art performance.

Introduction

Federated learning (FL) is a distributed computing paradigm that enables collaborative training across multiple distributed devices without requiring the upload of raw data, which has experienced remarkable growth in various domains such as healthcare (Wu et al. 2022; Guan et al. 2024), multimedia (Zhang, Liu, and Liu 2023; Li et al. 2024), Industrial Internet of Things (IIoT) (Boobalan et al. 2022; Ding et al. 2022) and so on. Unfortunately, traditional FL methods, such as FedAvg (McMahan et al. 2017), which only train a single global model, exhibit suboptimal end-of-training performance on severely non-independent and non-identically distributed (Non-IID) data (Liao et al. 2023; Huang et al. 2021), making it challenging to meet the personalized requirements of each client. Taking the development of tumor image detection in hospitals as an example of the application of FL (Jiang, Wang, and Dou 2022), it is evident that users of different demographic data, influenced by subtle variations in regions and eating habits, are likely to exhibit varying proportions of disease severity. Certain cases may predominantly occur within specific groups. In such scenarios, providing more personalized diagnostic predictions for each region becomes meaningful and necessary.

To tackle statistical heterogeneity and meet the personalized needs of clients in FL, personalized Federated Learning (PFL) methods have been proposed. Unlike traditional FL approaches that seek a globally optimal model with strong generalization by training and collaborating across distributed clients, PFL aims to train a set of personalized models to mitigate the impact of Non-IID across different clients. Recent studies in PFL fall into two main categories based on their underlying motivations: (1) Architecture-driven methods that aim to achieve personalization through customized model designs tailored to each client, including FedPer (Arivazhagan et al. 2019) and FedGH (Yi et al. 2023), (2) Similarity-based methods that focus on achieving personalization by modeling client relationships, including FedAMP (Huang et al. 2021) and FeSEM (Long et al. 2023).

The PFL methods in category (1) face challenges in determining the optimal architectural design. Furthermore, at the private parameter level, there is no direct benefit from other clients with similar data distribution, potentially overlooking valuable information from them. Meanwhile, the PFL methods in category (2) still have shortcomings. On the one hand, due to the modeling of user relationships, the personalization models of clients are easily influenced by other clients, making these methods sensitive to the poor data quality of clients. On the other hand, these methods primarily focus on client-level model aggregation, lacking fine-grained knowledge sharing relevant to local tasks. In addition, this process may introduce additional computational and communication costs for clients (Ghosh et al. 2022), particularly detrimental to edge devices with constrained resources.

Recent studies have revealed that the degradation in FL predominantly stems from classifier biases in clients' local models induced by Non-IID data (Li et al. 2023; Luo et al.

*Corresponding authors.

2021). In particular, it has been observed that the classifier layer exhibits higher biases compared to other layers (Luo et al. 2021). These classifier biases engender a vicious cycle, in which biased classifiers and misaligned features across clients reinforce each other (Zhou, Zhang, and Tsang 2023). Fig.1 illustrates a toy example showing the distance matrix of the classifier (a fully connected layer) parameters across all clients. The distribution of Non-IID labels is generated by pathological partitioning, where every four clients are assigned the same two classes of labels. It can be observed that clients with similar data distributions tend to exhibit lower pair-wise parameter distance (e.g., ℓ_2-norm distance) among clients' classifiers in the Non-IID data setting. This motivates us to exploit a more efficient mechanism to select the collaborators calibrated to each client (the paired clients corresponding to the dark region in Fig.1), without the need for prior knowledge of the number of groups, as required by most existing clustered federated learning methods (Ding et al. 2022; Ghosh et al. 2022; Long et al. 2023).

In order to address the aforementioned problems and enhance collaborative learning among clients in PFL, we introduce a novel PFL method called PFedCS. It adaptively performs clustering with each client as a cluster center based on the distances of classifier parameters between clients, thereby identifying appropriate collaborating clients with similar data distributions. Since the lower layers in deep neural networks (DNN) focus more on universal information compared to higher layers (Yu et al. 2018; Oh, Kim, and Yun 2022), a distance-constrained aggregate weight is designed for each client to generate customized classifiers with the help of collaborative clients to guide local training in PFedCS, while the lower layers of all clients are aggregated to extract global features. Subsequently, the customized classifier, after undergoing several rounds of local fine-tuning, serves as the teacher model to guide the training of the local classifier. Although the introduction of customized classifiers requires a slight computational cost for local fine-tuning, the search for collaborating clients and the aggregation process both occur on the central server, which has significantly greater computational resources than the clients. As a result, PFedCS does not significantly increase the computational and communication cost on the client side. In this paper, we focus on the scenario of *label distribution shift* in the Non-IID data setting. Extensive experimental results on various datasets have strongly validated the effectiveness of our proposed method. Our main contributions are summarized as follows:

- We observe that clients with similar data distributions in Non-IID scenarios tend to exhibit a smaller pair-wise parameter distance of the classifier parameters. Based on the insight and observation, the pair-wise distances of the classifier parameters can be computed to be aware of the similarity of their data distributions. This motivates us to encourage collaboration among clients with similar classifiers to avoid the negative effect of Non-IID data.
- We propose a novel PFL method called PFedCS that adaptively selects collaborators for each client based on the distances of the classifier parameter. Moreover, we design a distance-constrained aggregation method to generate customized classifiers to guide local training.
- We conduct extensive experiments over various datasets to validate the effectiveness of PFedCS, which outperforms twelve state-of-the-art methods by up to 4.07% in test accuracy. In addition, we empirically show that PFedCS exhibits superior performance with different numbers of clients.

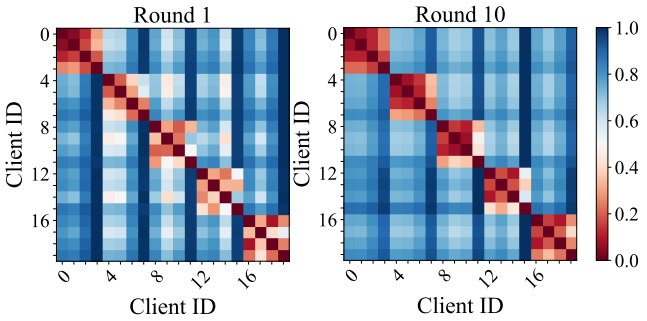

Figure 1: Motivation of PFedCS. It visualizes the normalized classifier distance matrices D^1 and D^{10} on CIFAR-10 across clients in different communication rounds.

Related Work

Federated Learning

Federated learning was first proposed by McMahan et al. (McMahan et al. 2017), also known as FedAvg. This approach aims to train a generalized global model by leveraging the local models of all clients. In the current landscape where privacy concerns are receiving increasing attention, the advantages of FL have become prominent. Consequently, a substantial body of research efforts have been dedicated to addressing the limitations of Federated Learning from diverse perspectives, including system heterogeneity (Xia et al. 2022; Wang et al. 2024), statistical heterogeneity (Qi et al. 2023; Mendieta et al. 2022), communication efficiency (Cheng et al. 2023; Wang et al. 2023), as well as security and privacy issues (Wu et al. 2021, 2023). Recently, numerous studies have demonstrated that although FedAvg performs well in the setting of independent and identically distributed (IID) data, its performance is considerably suboptimal in Non-IID data scenarios, and may even be inferior to the separate training performed by each client locally (Li et al. 2022; Jiang and Lin 2023). To address this problem, previous studies have attempted to enhance the robustness of the global model (Li et al. 2020; Huang et al. 2023). However, a single global model is challenging in meeting the personalized requirements of each client and fails to systematically address the challenges posed by data heterogeneity.

Personalized Federated Learning

Compared to the strategy of training a robust global model, Personalized Federated Learning emphasizes training multiple personalized models. From the perspective of learning personalized models, there are two mainstream methods in

PFL (Tan et al. 2022a; Huang et al. 2024): Personalized aggregation based on customized model architectures and personalization based on client similarity modeling.

Architecture-driven PFL methods aim to achieve personalization by providing custom model designs for each client. A representative category of methods involves parameter decoupling, such as FedPer (Arivazhagan et al. 2019), FedRep(Collins et al. 2021), FedBABU (Oh, Kim, and Yun 2022), FedPCL (Tan et al. 2022c) and FedGH (Yi et al. 2023), which decouple the local private model parameters from the global FL model parameters. The private parameters are retained on the client side. This type of method provides flexibility for each client's architectural design. However, at the private parameter level, there is no direct benefit from other clients with similar data distributions, potentially overlooking valuable information from other clients. Inspired by them, we have also adopted a design of model parameter decoupling, with the distinction that we apply different personalized knowledge-sharing mechanisms to different components of the parameters.

Similarity-based PFL methods achieve personalization by modeling relationships among clients, in which similar clients collaborate to enhance the learning of similar models (T Dinh, Tran, and Nguyen 2020; Li et al. 2021). For example, FedAMP (Huang et al. 2021) designs an attention-based mechanism that enhances collaboration among FL clients with similar data distributions. FedFomo (Zhang et al. 2021) updates the personalized models by a weighted combination of all clients' models based on the loss similarities. These methods focus mainly on pair-wise client relationships. There are also works that consider client relationships at the group level (Sattler, Müller, and Samek 2020; Long et al. 2023; Vahidian et al. 2023). However, determining the appropriate number of groups remains a challenging problem in the absence of prior knowledge about the number of distinct categories. Furthermore, these approaches directly aggregate the global model and lack fine-grained collaboration with local task-specific classifiers. In this work, we resort to the server to perform collaborative client selection and fine-grained collaboration without incurring additional communication costs.

Problem Settings

Following typical federated learning (McMahan et al. 2017; Li et al. 2020), where K is the number of clients in the federated learning system (indexed by k), $\mathbb{C}^t \subseteq \mathbb{C}$ is the subset of selected clients in round $t \in [1, T]$ and T is the total federated rounds. In addition, each client k has a private Non-IID dataset $\mathcal{D}_k = \left\{\boldsymbol{x}_i^k, y_i^k\right\}_{i=1}^{N_k}$, where $k \in \mathbb{C}$ and N_k is the number of samples in $\mathcal{D}_k$. We assume that these private datasets share the same feature space, but have different sample spaces. The entire dataset across all clients is denoted as $\mathcal{D} = \bigcup_{k=1}^{K} \{\mathcal{D}_k\}$. For traditional FL, the objective of the whole federated system is to obtain an optimal global model w_* as follows:

$$w_* = \min_{w} \sum_{k=1}^{K} p_k F_k(w) \tag{1}$$

where $p_k \geq 0$ is the weight of client k. Typically, in FedAvg (McMahan et al. 2017), p_k is set to $|\mathcal{D}_k|/|\mathcal{D}|$ and $\sum_{k=1}^{K} p_k = 1$. The local objective $F_k(\cdot)$ is often defined as the expected error over local dataset $\mathcal{D}_k$:

$$F_k(\cdot) = \mathbb{E}_{\left(\boldsymbol{x}_i^k, y_i^k\right) \sim \mathcal{D}_k} \left[\mathcal{L}_k(w; \boldsymbol{x}_i^k, y_i^k)\right] \tag{2}$$

where $\mathcal{L}_k$ is the loss function for the k-th client. For the local models, we can decompose them into two modules: A feature extractor f and a linear classifier g, i.e., $w = \{w_f, w_g\}$. For a given sample (x, y), the feature extractor $f : \mathcal{X} \to \mathcal{Z}$, parameterized by w_f, encodes the input sample into a d-dimension feature vector $z = f(x, w_f) \in \mathbb{R}^d$. The linear classifier $g : \mathcal{Z} \to \mathbb{R}^C$, parameterized by w_g, aggregates the information of the feature vector to produce a probability distribution $p^{w_g} = g(z, w_g)$ as the prediction result.

Due to the Non-IID data distribution across clients, the optimal global model obtained through training does not guarantee the best generalization performance across all clients. In this scenario, PFL allows for the coexistence of multiple models, enabling each client k to learn an optimal personalized model by aligning it with its local objectives:

$$\{w_*^{(1)}, \ldots, w_*^{(K)}\} = \min_{w_*^{(1)}, \ldots, w_*^{(K)}} \sum_{k=1}^{K} p_k F_k(w^{(k)}) \tag{3}$$

The objective of the PFL system is to minimize the overall empirical loss by considering collaborative learning and personalization to obtain the optimal local models $\{w_*^{(k)}\}$.

Methodology

Solution Overview. Before starting, we introduce the overview of the PFedCS method, which applies data distribution awareness and classifier collaboration to the federated scenario in detail. The overall framework of the proposed PFedCS is illustrated in Fig.2. The yellow and gray parts represent the global aggregation executed on the server and the local training on the clients, respectively. We divide the training process into 2 sub-stages. Assuming the stage 1 consists of β training rounds, in the $t \in [1, \min(\beta, T)]$ round, each client $k \in \mathbb{C}^t$ uploads the complete model $w^{(k)}$, while in the $t \in [\min(\beta, T), T]$ round of stage 2, clients retain their classifier $w_g^{(k)}$ locally. We outline the workflow of the federated training process in stage 1 as follows:

1. Each client k uploads the local model $w_{t-1}^{(k)} = \left\{w_{f,t-1}^{(k)}, w_{g,t-1}^{(k)}\right\}$ to the central server in round t.
2. The server first measures the similarity of the data distribution by calculating the distance matrix D^t. Then the server determines the collaborative clients $\overline{\mathbb{C}}_{k,c}^t$ for each client with a two-component GMM model and a dynamic threshold $\tau_{t,k}$. Finally, the server generates a set of customized classifiers $v_{g,t}^k$ with the help of $\overline{\mathbb{C}}_{k,c}^t$ for each client k and the same feature extractor $w_{f,t}$ for all clients (orange part in Fig. 2).
3. Active clients download the global feature extractor $w_{f,t}$, customized classifier $v_{g,t}^k$ from the server and replace the local feature extractor with $w_{f,t}$(step 1 in Fig. 2).

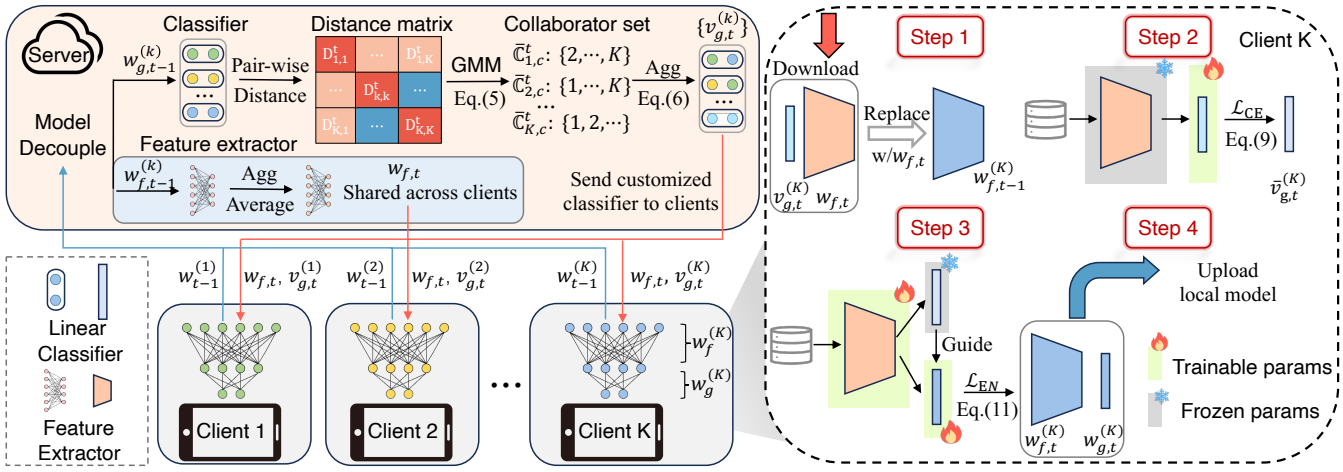

Figure 2: The overview of PFedCS. The left part presents the inference process. The right part shows the detailed process of local training for stage 1 of PFedCS. In stage 2, only the steps marked in blue in the left part are executed on the server.

4. Each client k freezes $w_{f,t}$ and fine-tunes $v_{g,t}^k$ over local data (step 2 in Fig. 2). After that, each client k unfreezes the local feature extractor and updates $w_{t-1}^{(k)}$ under the guidance of new $\overline{v}_{g,t}^k$ (step 3 in Fig. 2).

5. Each client k uploads the latest local model $w_t^{(k)}$ (step 4 in Fig. 2). The next round starts.

Different from stage 1, during stage 2, clients do not upload the classifier layer of their models to the server. Instead, they retain and train the classifiers locally. In other words, when the current round $t \geq \beta$, PFedCS degrades into FedPer(Arivazhagan et al. 2019) and only the blue steps in the left part of Fig. 2 are executed on the server. Next, we delve into in-depth discussions of stage 1 in PFedCS.

Adaptive Collaborator Selection Mechanism

During stage 1 of PFedCS, after receiving the latest local models, the server begins to identify collaborators for each client. At first, the server calculates pair-wise distances based on the classifier layer parameters of the client models. We assume that the classifier $w_g^{(k)}$ of client k is a linear transformation with weight $\varphi_k = [\varphi_{k,1}, \ldots, \varphi_{k,C}]$ and bias, followed by $Normalization$ and $Softmax$. In each round t, we employ a square matrix $D^t = [(D_{1,*}^t)^T, \ldots, (D_{K,*}^t)^T]^T \in \mathbb{R}^{K \times K}$ to record the distance between a pair of clients. For any clients i and j, the distance of classifier parameters between them is defined as $D_{i,j}^t = \frac{\|\varphi_{i,c} - \varphi_{j,c}\|_2^2}{\max(D_{i,*}^t)}$. Since the weights φ_k are typically initialized randomly, as the number of training rounds increases, this statistical information becomes more accurate. In PFedCS, the server builds D^t in every round. Based on D^t, each client k determines collaborative clients based on its distance from all other clients, i.e., $\widetilde{D_k^t} = D_{k,*}^t \setminus D_{k,k}^t$.

The current challenge lies in determining the collaborative clients. To select similar clients adaptively, we propose the utilization of clustering algorithms to group $\widetilde{D_k^t}$. The server computes a two-component Gaussian Mixture Model (GMM) on $\widetilde{D_k^t}$ for each client k. Then the set of clients $\mathbb{C}^t - \{k\}$ is grouped into two subsets: $\mathbb{C}_{k,c}^t$ (candidate clients) and $\mathbb{C}_{k,n}^t$ (non-candidate clients). Note that the group of clients with a lower mean distance serves as candidate clients. In this way, clients exhibiting similar data distributions are identified.

Distance-constrained Classifier Aggregation

At the beginning of FL training, it is difficult to identify clients with similar data distributions based on parameter distance due to the rapid fluctuations. Therefore, a better way is to aggregate the classifier parameters with more clients during early training. As training progresses, further refinement within candidate clients $\mathbb{C}_{k,c}^t$ becomes necessary.

To implement an effective refinement mechanism, we introduce a dynamic threshold, with its value negatively correlated with the number of training rounds. In round t, the server calculates a distance threshold for each client k:

$$\tau_{t,k} = \text{avg}(\widetilde{D_k^t}) + \frac{t}{\beta} \times (\min(\widetilde{D_k^t}) - \text{avg}(\widetilde{D_k^t})) \quad (4)$$

where the $\text{avg}(\cdot)$ and $\min(\cdot)$ function indicate the average and minimum value of the corresponding pair-wise distances within the set, respectively. As t increases, the threshold $\tau_{t,k}$ decreases, leading to a reduced number of clients selected for collaboration. Based on $\tau_{t,k}$, the server selects a subset of candidate clients from which we form the real collaborative client set:

$$\overline{\mathbb{C}}_{k,c}^t = \left\{ i \in \mathbb{C}_{k,c}^t | D_{k,i}^t \leq \tau_{t,k} \right\} \quad (5)$$

The server then leverages similar clients $\overline{\mathbb{C}}_{k,c}^t$ to generate a client-level customized classifier model for client k, which will be broadcast to client k for download in round t :

$$v_{g,t}^{(k)} = \sum p_{i,t} w_{g,t-1}^{(i)}, \quad i \in \left\{ \overline{\mathbb{C}}_{k,c}^t \cup k \right\} \quad (6)$$

where $v_{g,t}^{(k)}$ denotes the customized classifier downloaded by client k in round t. A smaller parameter distance between two clients indicates higher similarity in their data distributions, consequently enabling more effective mutual assistance during the aggregation. Therefore, it is necessary to increase the clients' weight with a smaller distance during the aggregation. Formally, we propose the distance-constrained classifier aggregation (DCA) and the weight is defined as:

$$p_{i,t} = \lambda \underbrace{\frac{D_{max} - D_{k,i}^t}{|\overline{\mathbb{C}}_{k,c}^t| \cdot (D_{max} - D_{avg})}}_{\textbf{Learn from more similar clients}} + (1-\lambda) \underbrace{\frac{N_i}{\sum_{j \in \overline{\mathbb{C}}_{k,c}^t} N_j}}_{\textbf{Learn from more data}} \tag{7}$$

$$D_{max} = \max(D_{k,j}^t), D_{avg} = \text{avg}(D_{k,j}^t), j \in \overline{\mathbb{C}}_{k,c}^t \tag{8}$$

where λ controls the importance of both similarity and the local data size when aggregating personalized classifiers. Finally, the server aggregates the received feature extractor as FedAvg, which helps all clients collaborate on facilitating the extraction of more generalizable features.

Customized Classifiers Guide Training

After obtaining the customized classifier $v_{g,t}$, a naive approach for clients to take advantage of its capabilities involves directly replacing their local classifiers $w_{g,t-1}$ with the new classifier $v_{g,t}$. However, the inference objectives of the server-generated customized classifier and the local classifier may be inconsistent, leading to oscillations in loss during local training, resulting in degraded performance.

Inspired by FedRod (Chen and Chao 2022), as a global model with stronger generalization capabilities can achieve a higher level of personalization after local adaptation, we propose to fine-tune the customized classifier v_g, familiarizing it with the local task and enabling personalization. In round t, after receiving new $w_{f,t}$ and $v_{g,t}^{(k)}$ from the server, the client k first updates the local feature extractor $w_{f,t-1}^{(k)}$ with $w_{f,t}$. Then, the client k freezes $w_{f,t-1}^{(k)}$ and fine-tunes $v_{g,t}^{(k)}$ for ρ epochs via Stochastic Gradient Descent (SGD) :

$$v_{g,t}^{(k)} \leftarrow v_{g,t}^{(k)} - \eta_v \nabla_{v_{g,t}^{(k)}} \mathcal{L}_{CE}\left(p^v; y\right), \tag{9}$$

$$p^v = g(f(x, w_{f,t-1}^{(k)}, v_{g,t}^{(k)}), (x,y) \sim \mathcal{D}_k \tag{10}$$

where η_v is the learning rate for fine-tuning customized classifier and $\mathcal{L}_{CE}$ is the cross-entropy loss between the output logits p^v and the true class label y. The objective of this step is to promote alignment between the extracted features and classifier vectors while enabling the biased classifiers to absorb prior knowledge from the local class distributions. After local fine-tuning, the customized classifier is updated to $\overline{v}_{g,t}^k$, which is used to guide the training of the local model, maximizing the benefits of the personalized classifier while minimizing disruptions to the local training. Specially, the client k unfreezes $w_{f,t-1}^{(k)}$ and freezes $\overline{v}_{g,t}^{(k)}$, then performs SGD to update $w_{t-1}^{(k)} = \left\{w_{f,t-1}^{(k)}, w_{g,t-1}^{(k)}\right\}$:

$$w_{t-1}^{(k)} \leftarrow w_{t-1}^{(k)} - \eta_w \nabla_{w_{t-1}^{(k)}} \mathcal{L}_{EN}\left(w_{t-1}^{(k)}, v_{g,t}^{(k)}; \mathcal{D}_k\right) \tag{11}$$

where η_w is the learning rate for local training. Our objective is to improve the performance on the local task, with the guidance provided by the personalized classifier $\overline{v}_{g,t}^k$. Following (Jin et al. 2023), we propose to use $\overline{v}_{g,t}^k$ as a teacher to guide the local classifier $w_{g,t}^{(k)}$ in assimilating the ensemble knowledge within the integrated classifier as follows:

$$\mathcal{L}_{EN} := \mathcal{L}_{CE}(p^w, y) + \mathcal{D}_{KL}(p^v || p^w) \tag{12}$$

$$\text{where} \quad p^w = g(f(x, w_{f,t}^{(k)}), w_{g,t}^{(k)}), (x,y) \sim \mathcal{D}_k \tag{13}$$

where the term $\mathcal{D}_{KL}(p^v||p^w)$ represents the Kullback-Leibler (KL) divergence between the output logits of w_g and v_g, whose objective is to guide the local model in assimilating the ensemble knowledge in the teacher model.

Experiments

This section presents the experiment setups, comparison with SOTA methods and ablation studies.

Experiment Setup

Datasets and Models. Our experiments are conducted on three public datasets: CIFAR-10 (Krizhevsky, Hinton et al. 2009), CIFAR-100 (Krizhevsky, Hinton et al. 2009), and Tiny-ImageNet (Chrabaszcz, Loshchilov, and Hutter 2017). All datasets are randomly divided into the training and test sets following a 3:1 split. We use the data partitioning methods in (Li et al. 2022) to simulate different label skews. Specifically, we try two types of Non-IID partition: (1) Pathological Non-IID (McMahan et al. 2017): we sample C classes for CIFAR-10/CIFAR-100/Tiny-ImageNet from 10/100/200 classes for each client, with disjoint data and different numbers of data samples. (2) Practical Non-IID (Zhang et al. 2023): We sample a proportion of samples of class j to client k with Dirichlet distribution, i.e., $p_{j,k} \sim Dir(\alpha)$ and smaller α leads to greater class imbalance. Following (McMahan et al. 2017; Dai et al. 2023), we consider a 4-layer CNN that consists of two convolutional layers and two fully connected layers for CIFAR-10 and CIFAR-100, and ResNet-18 (He et al. 2016) for Tiny-ImageNet, respectively. For all model decoupling methods, we use a linear layer as the classifier, while considering the remaining parts as the feature extractor. For each setting, clients' local training and test datasets are under the same distribution.

Compared methods. We compare our PFedCS with twelve state-of-the-art FL algorithms, including two traditional FL algorithms: The leading **FedAvg** (McMahan et al. 2017) and **FedProx** (Li et al. 2020), and ten PFL algorithms: **Per-FedAvg** (Fallah, Mokhtari, and Ozdaglar 2020), **FedPer** (Arivazhagan et al. 2019), **FedBABU** (Oh, Kim, and Yun 2022), **FedAMP** (Huang et al. 2021), **FedFomo** (Zhang et al. 2021), **FedProto** (Tan et al. 2022b), **FedRod** (Chen and Chao 2022), **FedGH** (Yi et al. 2023), **ClusterFL** (Sattler, Müller, and Samek 2020) and **FeSEM** (Long et al. 2023). Note that ClusterFL and FeSEM are specially designed for clustered FL. Following FedAMP (Huang et al. 2021), we report the mean top-1 accuracy by averaging the test accuracies over all clients with 3 trials.

Method	CIFAR-10			CIFAR-100			Tiny-Imagenet			Average
	$C=2$	$C=3$	$C=4$	$C=10$	$C=15$	$C=20$	$C=20$	$C=30$	$C=40$	
FedAvg	54.29	55.06	58.82	23.36	23.96	24.33	16.89	17.23	19.07	32.56
FedProx	54.25	55.00	58.67	23.39	24.05	24.20	16.74	17.53	18.73	32.51
Per-FedAvg	88.82	84.44	81.29	58.97	52.32	46.11	32.84	27.66	25.65	55.34
FedPer	90.06	84.66	80.75	60.85	53.29	46.38	41.21	35.29	32.51	58.33
FedBABU	88.37	83.53	80.39	60.55	53.90	46.72	42.86	37.39	36.04	58.86
FedAMP	89.25	83.18	78.13	60.80	52.10	44.61	39.54	32.37	28.97	56.55
FedFomo	90.48	85.15	81.80	60.55	54.13	48.35	36.51	31.05	30.14	57.57
FedProto	88.17	83.06	77.18	57.41	54.07	46.54	38.20	30.93	27.25	55.87
FedRod	90.24	85.12	81.12	57.73	51.73	46.44	38.87	33.71	31.70	57.41
FedGH	89.25	82.92	78.62	61.64	54.00	46.22	38.23	31.61	27.21	56.63
ClusterFL	89.72	83.96	80.94	62.01	56.06	47.85	38.07	37.36	36.19	59.13
FeSEM	89.29	83.14	78.10	60.91	52.21	44.73	39.32	32.58	28.75	56.56
PFedCS	**90.60**	**85.49**	**82.06**	**63.29**	**56.15**	**50.00**	**46.93**	**41.36**	**38.92**	**61.64**

Table 1: Comparison results in the pathological Non-IID setting on CIFAR-10, CIFAR-100, and Tiny-Imagenet.

Method	CIFAR-10			CIFAR-100			Tiny-Imagenet			Average
	$\alpha=0.01$	$\alpha=0.05$	$\alpha=0.1$	$\alpha=0.01$	$\alpha=0.05$	$\alpha=0.1$	$\alpha=0.01$	$\alpha=0.05$	$\alpha=0.1$	
FedAvg	39.53	44.69	50.17	22.30	25.85	27.37	13.86	14.89	15.23	28.21
FedProx	39.58	44.69	50.15	22.22	25.77	27.40	13.99	14.42	14.50	28.08
Per-FedAvg	96.60	92.04	88.61	65.75	53.54	46.97	46.88	36.56	29.67	61.85
FedPer	97.17	93.52	91.05	68.03	56.49	47.44	54.90	45.16	39.47	65.91
FedBABU	96.59	92.70	90.00	65.92	56.30	48.69	57.47	46.44	39.06	65.91
FedAMP	97.28	**93.76**	90.55	69.70	56.78	47.38	52.76	42.62	34.54	65.04
FedFomo	97.27	93.20	89.86	68.32	54.13	44.80	49.78	38.08	31.92	63.04
FedProto	96.29	92.84	87.98	66.00	54.34	45.67	49.63	39.82	32.62	62.80
FedRod	97.14	93.21	90.89	64.74	56.02	48.52	49.58	43.95	36.84	64.54
FedGH	96.11	83.92	89.32	69.47	56.60	47.78	54.57	40.68	30.22	63.19
ClusterFL	95.22	88.44	83.45	56.59	47.86	41.77	45.72	36.38	31.20	58.51
FeSEM	97.28	93.73	90.54	69.84	56.68	47.39	52.37	42.34	33.90	64.90
PFedCS	**97.30**	93.58	**91.56**	**70.20**	**59.39**	**50.60**	**60.26**	**49.55**	**43.22**	**68.41**

Table 2: Comparison results in the practical Non-IID setting on CIFAR-10, CIFAR-100, and Tiny-Imagenet.

Implementation details. We run the baselines in the settings suggested in the original papers. We adopt SGD optimizer and set the batch size to 100. Regarding the local learning rate η_w, we set $\eta_w = 0.005$ for 4-layer CNN and $\eta_w = 0.1$ for ResNet-18. Note that PFedCS introduces an additional learning rate η_v for fine-tuning customized classifier, we set $\eta_v = \eta_w$ by default. We run 200 federated rounds and set the number of local epochs to 5 to guarantee convergence. The number of clients is set to 20 and the client joining ratio is set to 1 by default. Unless specifically stated, the settings are shared for all experiments.

Comparisons with State-of-the-arts

Pathological Skew Settings. We report the test accuracy results in the pathological Non-IID data setting in Table 1. For the CIFAR-10, CIFAR-100 and Tiny-ImageNet datasets, we sample $C = \{2, 3, 4\}/\{10, 15, 20\}/\{20, 30, 40\}$ classes for each client, respectively. The optimal results are indicated in bold and the sub-optimal results are underlined. It is obvious that PFedCS performs best on three datasets and outperforms the best baseline by an average of 2.51%. Among the baselines, the traditional FL algorithms (FedAvg and FedProx) perform poorly because they only train a single global model for all clients, failing to meet the personalized requirements of different clients. The superiority of PFedCS can be attributed to the ability to adaptively identify clients with similar data distributions through the distance between their classifier parameters. It leverages the ensemble knowledge from collaborative clients to guide local model learning, thereby achieving superior performance across all datasets.

Practical Skew Settings. We also report the test accuracy results under the Dirichlet Non-IID data setting in Table 2, with α values of $\{0.01, 0.05, 0.1\}$ for all three datasets. Compared to baselines, PFedCS obtains the best performance in all conditions except that on CIFAR-10 with $\alpha = 0.05$, which is slightly inferior to FedAMP. We attribute the superior performance of FedAMP to simpler datasets. Moreover, PFedCS exhibits outstanding performance on the Tiny-ImageNet dataset, surpassing the sub-optimal method

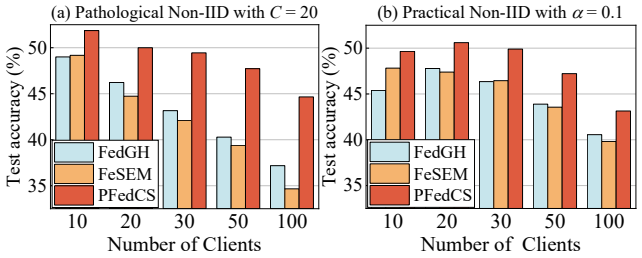

Figure 3: Results on CIFAR-100 versus numbers of clients.

by 2.50% in average accuracy across three different settings of α. This result demonstrates the effectiveness of PFedCS in adapting to Non-IID scenarios of varying complexity.

Varying Numbers of Clients. Following MOON (Li, He, and Song 2021), we split the CIFAR-100 dataset into $\{10, 20, 30, 50, 100\}$ sub-datasets to form the corresponding number of clients to present the effectiveness of PFedCS with different numbers of clients. The results of PFedCS and the other ten PFL baselines are shown in Fig.3. Unfortunately, as the number of clients increases, the average number of samples assigned to each client decreases, which leads to a performance drop across all methods. It can be found that PFedCS consistently outperforms other baselines with different numbers of clients, demonstrating the adaptability and scalability of PFedCS in heterogeneous data scenarios.

Methods/Non-IID	$C = 10$	$C = 15$	$C = 20$
FedPer	60.85	53.29	46.38
Select None	60.51	53.35	46.60
Select All	61.57	54.00	48.28
Random Selection	61.41	54.34	48.30
Adaptive Selection (Ours)	**63.29**	**56.15**	**50.00**

Table 3: The test accuracy (%) of PFedCS and its collaborator selection mechanism variants on CIFAR-100.

Ablation Studies

Effectiveness of Adaptive Selection Mechanism. Table 3 compares our adaptive selection mechanism (ASM) with other selection mechanisms on CIFAR-100 in the pathological Non-IID setting. To be specific, we introduce three collaborator selection mechanisms: (1) **Select None**: Each client forms a separate group, with only the feature extractor uploaded and aggregated on the server, thus degrading to FedPer (Arivazhagan et al. 2019); (2) **Select All**: Each client selects all other clients as collaborators; (3) **Random Selection**: Each client randomly selects a fixed number of clients as a group and we set it to 3. The results show that applying our adaptive selection mechanism improves performance across Non-IID scenarios by an average of 1.74%. This indicates collaboration among clients with similar classifier parameters can yield significant performance benefits. The results demonstrate that better model performance can be achieved by measuring the distance between client classifier parameters to select clients with similar distributions for collaboration.

Dataset	KM	HIER	FIX	PFedCS
CIFAR-10	90.50	90.51	90.48	**90.60**
CIFAR-100	62.65	62.67	61.49	**63.29**
Tiny-ImageNet	46.28	46.68	40.23	**46.93**

Table 4: The test accuracy (%) of PFedCS and its clustering method variants in the pathological Non-IID setting.

Different Grouping Strategies in ASM. To explore the effects of clustering methods in dividing $\mathbb{C}_{k,c}^t$ from $\mathbb{C}^t$, we replace the GMM with K-Means and hierarchical clustering, denoted by "KM" and "HIER". In addition, we set a fixed threshold (set to 0.5) to replace the clustering algorithm, denoted by "FIX". As Table 4 shows, the performance differences when using different clustering methods are small, but the results obtained with the fixed threshold perform poorly, which decrease by 6.70% on Tiny-ImageNet. It's clear that using clustering methods in PFedCS is effective and PFedCS shows robustness to the clustering algorithm chosen.

	$\rho = 0$	$\rho = 1$	$\rho = 2$	$\rho = 3$	$\rho = 5$	$\rho = 10$
Acc.	53.58	59.39	**59.51**	59.41	59.46	59.16

Table 5: The test accuracy (%) on CIFAR-100 in the practical setting ($\alpha = 0.05$) with different fine-tuning epochs ρ.

Effects of Fine-tuning Epochs ρ. Here, we study the effects of ρ on test accuracy. The results of PFedCS by varying ρ are shown in Table 5. It can be seen that the accuracy first increases from $\rho = 0$ to $\rho = 1$, then the accuracy maintains stable from $\rho = 1$ to $\rho = 5$ but decreases from $\rho = 5$ to $\rho = 10$. A larger ρ leads to better global knowledge absorption, but it also incurs higher computational costs and may result in the catastrophic forgetting problem (Shenaj et al. 2023; Huang, Ye, and Du 2022) in neural networks. Therefore, we adopt $\rho = 1$ by default to achieve a trade-off between computational cost and performance.

Conclusion

In this work, we propose a novel PFL method dubbed PFedCS, to address the limitation of existing FL methods in lacking fine-grained collaboration among clients with similar classifiers in data heterogeneous scenarios. The key insight is to leverage the distances between classifier parameters of clients to perceive the similarities in data distributions and promote collaboration among similar clients. Through iterative distance measurement, collaborator selection, and distance-constrained aggregation, PFedCS can adaptively identify clients with similar data distributions and generate customized classifiers to guide local training. Extensive experiments on various datasets demonstrate that PFedCS achieves state-of-the-art performance.

Acknowledgments

This research is supported by the National Natural Science Foundation of China No.92267104 and Dou Wanchun Expert Workstation of Yunnan Province No.202105AF150013. The authors wish to acknowledge Dr. Fei Dai, Professor of Southwest Forestry University, for his help in interpreting the significance of the results of this study.

References

Arivazhagan, M. G.; Aggarwal, V.; Singh, A. K.; and Choudhary, S. 2019. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*.

Boobalan, P.; Ramu, S. P.; Pham, Q.-V.; Dev, K.; Pandya, S.; Maddikunta, P. K. R.; Gadekallu, T. R.; and Huynh-The, T. 2022. Fusion of federated learning and industrial Internet of Things: A survey. *Computer Networks*, 212: 109048.

Chen, H.-Y.; and Chao, W.-L. 2022. On Bridging Generic and Personalized Federated Learning for Image Classification. In *International Conference on Learning Representations*.

Cheng, Z.; Xia, X.; Liwang, M.; Fan, X.; Sun, Y.; Wang, X.; and Huang, L. 2023. CHEESE: Distributed Clustering-Based Hybrid Federated Split Learning Over Edge Networks. *IEEE Transactions on Parallel and Distributed Systems*.

Chrabaszcz, P.; Loshchilov, I.; and Hutter, F. 2017. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*.

Collins, L.; Hassani, H.; Mokhtari, A.; and Shakkottai, S. 2021. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, 2089–2099. PMLR.

Dai, Y.; Chen, Z.; Li, J.; Heinecke, S.; Sun, L.; and Xu, R. 2023. Tackling data heterogeneity in federated learning with class prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 7314–7322.

Ding, Y.; Wu, X.; Li, Z.; Wu, Z.; Tan, S.; Xu, Q.; Pan, W.; and Yang, Q. 2022. An efficient industrial federated learning framework for AIoT: a face recognition application. *arXiv preprint arXiv:2206.13398*.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33: 3557–3568.

Ghosh, A.; Chung, J.; Yin, D.; and Ramchandran, K. 2022. An efficient framework for clustered federated learning. *IEEE Transactions on Information Theory*, 68(12): 8076–8091.

Guan, H.; Yap, P.-T.; Bozoki, A.; and Liu, M. 2024. Federated learning for medical image analysis: A survey. *Pattern Recognition*, 110424.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, W.; Ye, M.; and Du, B. 2022. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10143–10153.

Huang, W.; Ye, M.; Shi, Z.; Li, H.; and Du, B. 2023. Rethinking federated learning with domain shift: A prototype view. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16312–16322. IEEE.

Huang, W.; Ye, M.; Shi, Z.; Wan, G.; Li, H.; Du, B.; and Yang, Q. 2024. Federated learning for generalization, robustness, fairness: A survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2021. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 7865–7873.

Jiang, L.; and Lin, T. 2023. Test-Time Robust Personalization for Federated Learning. In *The Eleventh International Conference on Learning Representations*.

Jiang, M.; Wang, Z.; and Dou, Q. 2022. Harmofl: Harmonizing local and global drifts in federated learning on heterogeneous medical images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 1087–1095.

Jin, H.; Bai, D.; Yao, D.; Dai, Y.; Gu, L.; Yu, C.; and Sun, L. 2023. Personalized Edge Intelligence via Federated Self-Knowledge Distillation. *IEEE Transactions on Parallel and Distributed Systems*, 34(2): 567–580.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, G.; Ding, X.; Yuan, L.; Zhang, L.; and Rong, Q. 2024. Towards Resource-Efficient and Secure Federated Multimedia Recommendation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5515–5519. IEEE.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2022. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 965–978. IEEE.

Li, Q.; He, B.; and Song, D. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10713–10722.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, 6357–6368. PMLR.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2: 429–450.

Li, Z.; Shang, X.; He, R.; Lin, T.; and Wu, C. 2023. No Fear of Classifier Biases: Neural Collapse Inspired Federated Learning with Synthetic and Fixed Classifier. *arXiv preprint arXiv:2303.10058*.

Liao, X.; Liu, W.; Chen, C.; Zhou, P.; Zhu, H.; Tan, Y.; Wang, J.; and Qi, Y. 2023. HyperFed: Hyperbolic Prototypes Exploration with Consistent Aggregation for Non-IID Data in Federated Learning. *arXiv preprint arXiv:2307.14384*.

Long, G.; Xie, M.; Shen, T.; Zhou, T.; Wang, X.; and Jiang, J. 2023. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1): 481–500.

Luo, M.; Chen, F.; Hu, D.; Zhang, Y.; Liang, J.; and Feng, J. 2021. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34: 5972–5984.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Mendieta, M.; Yang, T.; Wang, P.; Lee, M.; Ding, Z.; and Chen, C. 2022. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8397–8406.

Oh, J.; Kim, S.; and Yun, S.-Y. 2022. FedBABU: Toward Enhanced Representation for Federated Image Classification. In *International Conference on Learning Representations*.

Qi, T.; Wu, F.; Lyu, L.; Huang, Y.; and Xie, X. 2023. Fedsampling: A better sampling strategy for federated learning. *arXiv preprint arXiv:2306.14245*.

Sattler, F.; Müller, K.-R.; and Samek, W. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8): 3710–3722.

Shenaj, D.; Toldo, M.; Rigon, A.; and Zanuttigh, P. 2023. Asynchronous federated continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5054–5062.

T Dinh, C.; Tran, N.; and Nguyen, J. 2020. Personalized federated learning with moreau envelopes. *Advances in neural information processing systems*, 33: 21394–21405.

Tan, A. Z.; Yu, H.; Cui, L.; and Yang, Q. 2022a. Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12): 9587–9603.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022b. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8432–8440.

Tan, Y.; Long, G.; Ma, J.; Liu, L.; Zhou, T.; and Jiang, J. 2022c. Federated learning from pre-trained models: A contrastive learning approach. *Advances in neural information processing systems*, 35: 19332–19344.

Vahidian, S.; Morafah, M.; Wang, W.; Kungurtsev, V.; Chen, C.; Shah, M.; and Lin, B. 2023. Efficient distribution similarity identification in clustered federated learning via principal angles between client data subspaces. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 10043–10052.

Wang, H.; Jia, Y.; Zhang, M.; Hu, Q.; Ren, H.; Sun, P.; Wen, Y.; and Zhang, T. 2024. FedDSE: Distribution-aware Submodel Extraction for Federated Learning over Resource-constrained Devices. In *Proceedings of the ACM on Web Conference 2024*, 2902–2913.

Wang, K.; He, Q.; Chen, F.; Jin, H.; and Yang, Y. 2023. FedEdge: Accelerating Edge-Assisted Federated Learning. In *Proceedings of the ACM Web Conference 2023*, 2895–2904.

Wu, Q.; Chen, X.; Zhou, Z.; and Zhang, J. 2022. FedHome: Cloud-Edge Based Personalized Federated Learning for In-Home Health Monitoring. *IEEE Trans. Mob. Comput.*, 21(8): 2818–2832.

Wu, S.; Zhang, G.; Dai, F.; Liu, B.; and Dou, W. 2023. An edge-assisted federated contrastive learning method with local intrinsic dimensionality in noisy label environment. *Software: Practice and Experience*.

Wu, Y.; Kang, Y.; Luo, J.; He, Y.; and Yang, Q. 2021. Fedcg: Leverage conditional gan for protecting privacy and maintaining competitive performance in federated learning. *arXiv preprint arXiv:2111.08211*.

Xia, J.; Liu, T.; Ling, Z.; Wang, T.; Fu, X.; and Chen, M. 2022. PervasiveFL: Pervasive federated learning for heterogeneous IoT systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11): 4100–4111.

Yi, L.; Wang, G.; Liu, X.; Shi, Z.; and Yu, H. 2023. Fedgh: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM International Conference on Multimedia*, 8686–8696.

Yu, F.; Wang, D.; Shelhamer, E.; and Darrell, T. 2018. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2403–2412.

Zhang, J.; Hua, Y.; Wang, H.; Song, T.; Xue, Z.; Ma, R.; Cao, J.; and Guan, H. 2023. GPFL: Simultaneously Learning Global and Personalized Feature Information for Personalized Federated Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5041–5051.

Zhang, M.; Sapra, K.; Fidler, S.; Yeung, S.; and Alvarez, J. M. 2021. Personalized Federated Learning with First Order Model Optimization. In *International Conference on Learning Representations*.

Zhang, Y.; Liu, L.; and Liu, L. 2023. Cuing without sharing: A federated cued speech recognition framework via mutual knowledge distillation. In *Proceedings of the 31st ACM International Conference on Multimedia*, 8781–8789.

Zhou, T.; Zhang, J.; and Tsang, D. H. 2023. FedFA: Federated Learning with Feature Anchors to Align Features and Classifiers for Heterogeneous Data. *IEEE Transactions on Mobile Computing*.