Dual Calibration-based Personalised Federated Learning

Xiaoli Tang[1] , **Han Yu**[1] , **Run Tang**[2] , **Chao Ren**[1] , **Anran Li**[1] and **Xiaoxiao Li**[3]
[1]College of Computing and Data Science, Nanyang Technological University, Singapore
[2]South China University of Technology, China
[3]Department of Electrical and Computer Engineering, The University of British Columbia, Canada
{xiaoli001, han.yu}@ntu.edu.sg, se_tangr@mail.scut.edu.cn, renchao1995@hotmail.com,
anran.li@ntu.edu.sg, xiaoxiao.li@ece.ubc.ca

Abstract

Personalized federated learning (PFL) is designed for scenarios with non-independent and identically distributed (non-IID) client data. Existing model mixup-based methods, one of the main approaches of PFL, can only extract either global or personalized features during training, thereby limiting effective knowledge sharing among clients. To address this limitation, we propose the Dual Calibration-based PFL (DC-PFL). It divides local models into a heterogeneous feature extractor and a homogeneous classifier. The FL server utilizes mean and covariance representations from clients' feature extractors to train a global generalized classifier, facilitating information exchange while preserving privacy. To enhance personalization and convergence, we design a feature extractor-level calibration method with an auxiliary loss for local models to refine feature extractors using global knowledge. Furthermore, DC-PFL refines the global classifier through the global classifier-level calibration, utilizing sample representations derived from an approximate Gaussian distribution model specific to each class. This method precludes the need to transmit original data representations, further enhancing privacy preservation. Extensive experiments on widely used benchmark datasets demonstrate that DC-PFL outperforms eight state-of-the-art methods, surpassing the best-performing baseline by 1.22% and 9.22% in terms of accuracy on datasets CIFAR-10 and CIFAR-100, respectively.

1 Introduction

As societies become increasingly concerned about data privacy protection when build artificial intelligence (AI) applications, federated learning (FL) [Zhuang *et al.*, 2023; Liu *et al.*, 2024] has emerged as a promising solution. It allows multiple data owners (a.k.a., FL clients) to collaboratively train models without exposing potentially sensitive local data. In a typical FL system, a central FL server coordinates multiple FL clients to perform collaborative model training. During each communication round, the server broadcasts the current global FL model to participating clients. Each client performs further training on this model using its private local data. The locally trained models are then uploaded to the FL server, which aggregates them to produce an updated global model. This iterative process continues until the FL model converges.

In practice, FL faces various types of heterogeneity issues which makes the aforementioned traditional FL paradigm unsuitable [Tan *et al.*, 2024]. To address these challenges, the field of personalized federated learning (PFL) [Tan *et al.*, 2022a] has emerged. The goal of PFL is to achieve collaborative learning and training reasonable personalized models for clients participated. Model mixup-based PFL methods [Jang *et al.*, 2022; Liang *et al.*, 2020; Collins *et al.*, 2021; Yi *et al.*, 2023] have gained significant research attention due to their ability to strike a balance between acceptable computational overhead and model performance without reliance on public datasets. They typically decompose the local models of clients into two distinct components: 1) a feature extractor, and 2) a classifier. The feature extractor transforms raw input data into latent space representations, while the classifier translates these representations into categorical vectors. During FL model training, the input into the feature extractor consists of both the global and the local feature information. Existing model mixup-based PFL methods can only extract either global feature information [Pillutla *et al.*, 2022; Oh *et al.*, 2022; Chen *et al.*, 2021; Collins *et al.*, 2021] or local feature information [Liu *et al.*, 2022; Jang *et al.*, 2022; Yi *et al.*, 2023]. This limits effective knowledge sharing among clients, which negatively impact the performance of resulting PFL models.

To deal with this issue, we propose the Dual Calibration-based Personalised Federated Learning (DC-PFL) approach. Under DC-PFL, clients' local models are decomposed into a heterogeneous feature extractor and a homogeneous classifier. Throughout the training process, the FL server utilizes mean and covariance representations extracted from clients' feature extractors to train a global generalized classifier shared across all the clients. This updated global classifier captures knowledge spanning all classes and clients, enabling knowledge exchange among diverse client models through a shared generalized global prediction classifier. These two operations facilitate the collaborative learning of DC-PFL.

To enhance the personalization of each client's model and improve model convergence, DC-PFL is incorporated with

a feature extractor-level calibration to train the personalised feature extractor by leveraging an auxiliary loss. This loss guides local models to refine their personalised feature extractors by effectively leveraging global knowledge. Moreover, DC-PFL performs global classifier-level calibration, which fine-tunes the global classifier using virtual representations and their corresponding labels. These virtual samples are derived from an approximate Gaussian distribution model tailored to each class, constructed using the mean and covariance class-specific representations. Importantly, this calibration step precludes the need to transmit original data representations, effectively addressing privacy concerns.

Through DC-PFL, clients benefit from personalized heterogeneous local models tailored to their unique data distributions, system resources, and model structures. The central FL server facilitates the dissemination of global knowledge among clients, leading to enhanced model performance. In addition, the virtual representations and auxiliary loss for feature extractors enable accurate model training, while minimizing the risk of privacy exposure. Extensive experiments on two widely used benchmark datasets demonstrate that DC-PFL is significantly more advantageous compared to eight state-of-the-art approaches, outperforming the best-performing baseline by 1.22% and 9.22% on average in terms of accuracy on CIFAR-10 and CIFAR-100, respectively.

2 Related Work

This paper explores PFL with the model heterogeneity [Yi *et al.*, 2023]. This domain has seen advancements along three main lines of work: 1) knowledge distillation-based PFL, 2) mutual learning-based PFL, and 3) model mixup-based PFL.

Knowledge Distillation-based PFL: Some knowledge distillation-based PFL approaches generally rely on public datasets for knowledge integration [Itahara *et al.*, 2021; Sattler *et al.*, 2021b; Makhija *et al.*, 2022; Chang *et al.*, 2019; Lin *et al.*, 2020; Huang *et al.*, 2022a; Li and Wang, 2019; Huang *et al.*, 2022b; Sattler *et al.*, 2021a; Li *et al.*, 2021; Cho *et al.*, 2022; Yu *et al.*, 2022; Cheng *et al.*, 2021]. However, finding suitable public datasets with similar distributions to local private data is not always guaranteed, limiting practical applicability. Alternatively, some methods operate independently of public datasets. For instance, [Zhang *et al.*, 2022; Zhu *et al.*, 2021] introduce a generator to produce public shared datasets or local representations. Nonetheless, the time-intensive iterative training for the generator reduces the overall computation efficiency of FL.

Mutual Learning-based PFL: In [Shen *et al.*, 2020; Wu *et al.*, 2022], each client needs to train a large heterogeneous model and a small homogeneous model simultaneously through mutual learning. The large model undergoes only local training, whereas the small model is sent to the FL server for aggregation. However, training a small homogeneous model on top of the heterogeneous large model increases the local computation costs for FL clients, which can be a significant burden for resource-constrained devices. While the mutual learning strategy facilitates information exchange among large heterogeneous models, the additional computation and communication overheads negatively impact the efficiency and scalability of such PFL approaches.

Model Mixup-based PFL: In [Pillutla *et al.*, 2022; Collins *et al.*, 2021; Oh *et al.*, 2022; Chen *et al.*, 2021], a homogeneous feature extractor is shared across all FL clients for server aggregation to improve generalization, while the classifier is heterogeneous for personalized classification tasks. However, a limitation is the larger parameter volume of the feature extractor than the classifier, restricting achievable model heterogeneity. In contrast, [Jang *et al.*, 2022; Liu *et al.*, 2022; Yi *et al.*, 2023; Liang *et al.*, 2020] use heterogeneous feature extractors and homogeneous classifiers, enabling higher degrees of model heterogeneity.

Our DC-PFL falls under the model mixup-based PFL category with heterogeneous feature extractors and homogeneous classifiers. However, different from existing methods, which only focus on either global feature information extraction or local feature information extraction, DC-PFL leverages the dual calibration to facilitate effective knowledge sharing among clients while enhancing the personalisation of each client's model, improving the performance of the resulting PFL models.

3 Preliminaries

An Overview of Federated Learning: In the FL system of interest, there are K FL clients and a central FL server. Each client $k \in [K]$ possesses a private dataset $D_k = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|D_k|}$, and the combined dataset is denoted as $D = \cup_{k=1}^{K} D_k$. The dataset contains C classes, and each sample in D is denoted as $(\boldsymbol{x}, y) \in \mathcal{X} \times [C]$, where $\mathcal{X}$ denotes the input space, $\boldsymbol{x}$ represents the input data (e.g., an image) while y is the corresponding label. Furthermore, we represent the collection of samples with the actual label $c \in [C]$ from client k as $D_k^c = \{(\boldsymbol{x}, y) \in D_k : y = c\}$.

The FL process operates through communication between the central server and clients in a round-based manner. In communication round t, the central server broadcasts the current global model parameter $\boldsymbol{w}^{t-1}$ to a selected set of clients, denoted as S^t. Upon receiving the global model $\boldsymbol{w}^{t-1}$, each selected client $k \in S^t$ performs a local update to obtain $\boldsymbol{w}_k^t$ based on its private data, guided by the following objective function: $\operatorname{argmin}_{\boldsymbol{w}_k^t} \mathbb{E}_{(\boldsymbol{x},y)\sim D_k}[\mathcal{L}(\boldsymbol{w}_k^t; (\boldsymbol{x}, y)]$, where $\mathcal{L}(\cdot)$ denotes the loss function, contingent on the current global model parameters $\boldsymbol{w}^{t-1}$ and the FL model aggregation algorithm. For example, FedAvg [McMahan *et al.*, 2017] calculates $\boldsymbol{w}_k^t$ by employing SGD [Robbins and Monro, 1951] on D_k for a certain number of epochs using the cross-entropy loss. The parameter set is initialized to $\boldsymbol{w}^{t-1}$. At the end of round t, each selected client $k \in S^t$ sends its optimized parameter $\boldsymbol{w}_k^t$ to the central server. The global parameter is then updated by aggregating these diverse parameters, i.e., $\boldsymbol{w}^t = \sum_{k\in S^t} p_k \boldsymbol{w}_k^t$, where $p_k = \frac{|D_k|}{\sum_{k'\in S^t} |D_{k'}|}$. The aforementioned steps are iterated until the global model converges. The objective of FedAvg is to minimize the average loss of the final global model $\boldsymbol{w}$ based on all clients' data: $\operatorname{argmin}_{\boldsymbol{w}} = \sum_{k\in[K]} \frac{|D_k|}{|D|} \mathcal{L}(\boldsymbol{w})$.

In traditional FL settings, all clients' local models must have the same structures, necessitating homogeneity, due to

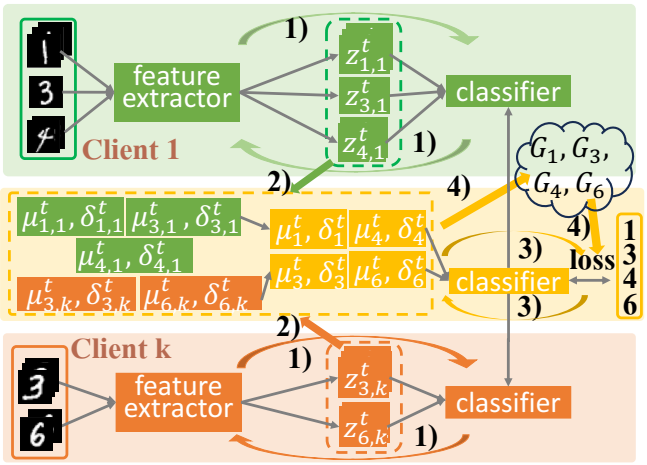

Figure 1: The workflow of DC-PFL. The colored arrows with numbers are the main steps of DC-PFL in each training round: 1) local model training & calibration, 2) class feature representation calculation, 3) global classifier training, and 4) global classifier calibration.

the requirement of averaging the received local models.

Problem Definition: This paper studies PFL across K clients, each having heterogeneous local models, but all involved in the same supervised classification tasks. We decompose the local model (which in this paper pertains to the classification model) of each client into a personalized and heterogeneous feature extractor alongside a global and homogeneous classifier. For a given sample $(\boldsymbol{x}, y) \in D_k$, the feature extractor $f_{\boldsymbol{\theta}_k} : \mathcal{X} \to \mathcal{Z}$ is governed by parameter θ_k, transforming input image $\boldsymbol{x}$ into feature space $\mathcal{Z}$, denoted as feature vector $\boldsymbol{z} = f_{\boldsymbol{\theta}_k}(\boldsymbol{x}) \in \mathbb{R}^d$. Subsequently, the classifier $g_{\boldsymbol{\varphi}} : \mathcal{Z} \to \mathbb{R}^C$, with parameters $\boldsymbol{\varphi}$, generates a probability distribution $g_{\boldsymbol{\varphi}}(\boldsymbol{z})$ that functions as the prediction for the provided input image $\boldsymbol{x}$. Consequently, the parameter of the local classification model of client k can be denoted as $\boldsymbol{w}_k = (\boldsymbol{\theta}_k, \boldsymbol{\varphi})$. The central server solely possesses the homogeneous classifier $g_{\boldsymbol{\varphi}}$. The primary objective is to minimize the sum of the losses of the local heterogeneous classification models of all K clients, which can be defined as:

$$\underset{\boldsymbol{w}_k, k\in[K]}{\operatorname{argmin}} = \sum_{k\in[K]} \mathcal{L}(\boldsymbol{w}_k). \tag{1}$$

4 The Proposed DC-PFL Approach

Figure 1 illustrates the workflow of DC-PFL during each training round t, which involves four main steps: 1) Local Model Training and Calibration, 2) Class Feature Representation Calculation, 3) Global Classifier Training, and 4) Global Classifier Calibration.

These steps are executed iteratively until the local classification models of all clients converge, ensuring that every client's local knowledge is integrated effectively into the global model. Once the FL process concludes, the personalized heterogeneous local models are ready for inference, enabling efficient and accurate predictions on diverse client datasets. In the following sections, we introduce each of these four steps in detail, elaborating on their significance and contributions to the overall DC-PFL approach.

4.1 Local Model Training & Calibration

Similar to the conventional FL training process, in each training round t, after receiving the global classifier $\boldsymbol{\varphi}^t$ from the central server, each client $k \in S^t$ updates its local classification model $\boldsymbol{w}_k^{t-1} = (\boldsymbol{\theta}_k^{t-1}, \boldsymbol{\varphi}^{t-1})$ to $\hat{\boldsymbol{w}}_k^t = (\hat{\boldsymbol{\theta}}_k^t, \hat{\boldsymbol{\varphi}}^t)$, where $\hat{\boldsymbol{\theta}}_k^t \leftarrow \boldsymbol{\theta}_k^{t-1}$ and $\hat{\boldsymbol{\varphi}}^t \leftarrow \boldsymbol{\varphi}^{t-1}$. Then, client k proceeds to calculate the supervised loss based on its local data D_k as:

$$\mathcal{L}_{sup}(\hat{\boldsymbol{w}}_k^t; D_k) = \frac{1}{|D_k|} \sum_{(\boldsymbol{x},y)\in D_k} \mathcal{L}_{CE}(g_{\hat{\boldsymbol{\varphi}}^t}(f_{\hat{\boldsymbol{\theta}}_k^t}(\boldsymbol{x})), y), \tag{2}$$

where $\mathcal{L}_{CE}(\cdot)$ is the cross-entropy loss function.

In addition to the supervised loss, we incorporate an auxiliary loss for client k to facilitate the learning of its local classification model, which is referred to as the feature extractor-level calibration. This process involves the central server broadcasting not only the global classifier $\boldsymbol{\varphi}^t$ but also the global mean $\boldsymbol{\mu}_c^t$ of all images belonging to the same class $c \in [C]$ to the selected clients in the current iteration denoted as S^t. Subsequently, client k computes the auxiliary loss as:

$$\mathcal{L}_{kd}(\hat{\boldsymbol{w}}_k^t; D_k) = \frac{1}{|D_k|} \sum_{(\boldsymbol{x},y)\in D_k} ||f_{\hat{\boldsymbol{\theta}}_k^t}(\boldsymbol{x}) - \boldsymbol{\mu}_y^t||_2, \tag{3}$$

where $||\cdot||_2$ is the Euclidean norm.

The auxiliary loss in Eq. (3) serves the purpose of encouraging the features extracted by $\hat{\boldsymbol{\theta}}_k^t$ from data belonging to a specific class to resemble the given global representations, thereby facilitating the training of local feature extractor $\hat{\boldsymbol{\theta}}_k^t$.

The local loss of client k is then formulated as the combination of two components: the supervised loss defined in Eq. (2) and the auxiliary loss defined in Eq. (3). The resulting local loss function with hyper-parameter λ is expressed:

$$\mathcal{L}(\hat{\boldsymbol{w}}_k^t; D_k) = \mathcal{L}_{sup}(\hat{\boldsymbol{w}}_k^t; D_k) + \lambda \mathcal{L}_{kd}(\hat{\boldsymbol{w}}_k^t; D_k). \tag{4}$$

After formulating the local total loss, client k proceeds with local training and updates its local model parameters $\hat{\boldsymbol{w}}_k^t = (\hat{\boldsymbol{\theta}}_k^t, \hat{\boldsymbol{\varphi}}^t)$ to $\boldsymbol{w}_k^t = (\boldsymbol{\theta}_k^t, \boldsymbol{\varphi}^t)$ using gradient descent:

$$\boldsymbol{w}_k^t \leftarrow \hat{\boldsymbol{w}}_k^t - \eta \nabla \mathcal{L}(\hat{\boldsymbol{w}}_k^t; D_k), \tag{5}$$

where η denotes the learning rate.

By incorporating the auxiliary loss along with the supervised loss and performing local training with appropriate parameter updates, each client k actively contributes to the overall federated learning process while also benefiting from the shared global knowledge. This cooperative learning approach helps in achieving superior performance and convergence of the heterogeneous local models.

4.2 Class Feature Representation Calculation

After updating their local classification models, each client k generates features $\boldsymbol{z}_{k,i}^t = f_{\boldsymbol{\theta}_k^t}(\boldsymbol{x}_i)$ for each image $\boldsymbol{x}_i$ within its local dataset. Subsequently, the client computes the local mean $\boldsymbol{\mu}_{c,k}^t$ and covariance $\boldsymbol{\Sigma}_{c,k}^t$ for each class $c \in [C]$:

$$\boldsymbol{\mu}_{c,k}^t = \frac{1}{|D_k^c|} \sum_{(\boldsymbol{x}_i,y_i)\in D_k^c} \boldsymbol{z}_{k,i}^t, \tag{6}$$

$$\boldsymbol{\Sigma}_{c,k}^t = \frac{1}{|D_k^c|-1} \sum_{(\boldsymbol{x}_i, y_i) \in D_k^c} (\boldsymbol{z}_{k,i}^t - \boldsymbol{\mu}_{c,k}^t)(\boldsymbol{z}_{k,i}^t - \boldsymbol{\mu}_{c,k}^t)^T, \quad (7)$$

where D_k^c is the set of samples on client k that share the common ground-truth label c. The local mean $\boldsymbol{\mu}_{c,k}^t$ represents the average feature representation of class c within the local dataset of client k, while the local covariance $\boldsymbol{\Sigma}_{c,k}^t$ quantifies the dispersion of the features around the mean for that class. Client k then uploads the class-wise representations $\{(\boldsymbol{\mu}_{c,k}^t, \boldsymbol{\Sigma}_{c,k}^t)\}_{c\in[C]}$, along with their corresponding labels c to the central server. This information sharing facilitates effective global classifier calibration by the central server.

As highlighted in [Tan *et al.*, 2022b], the representations refer to high-level features extracted from the data. Therefore, inferring the original data from solely the extracted representations, without access to the parameters of the feature extractors, becomes challenging. In DC-PFL, client k only uploads class-wise representation means and covariances $\{(\boldsymbol{\mu}_{c,k}^t, \boldsymbol{\Sigma}_{c,k}^t)\}_{c\in[C]}$, significantly reducing the risk of privacy leakage even further. This privacy-preserving feature sharing enhances the overall security and confidentiality of the FL process, making it suitable for sensitive or private data scenarios.

4.3 Global Classifier Training

The central server incorporates all the uploaded class-wise representation means and covariances $\{(\boldsymbol{\mu}_{c,k}^t, \boldsymbol{\Sigma}_{c,k}^t)\}_{c\in[C]}$ received from each selected client at round t into the global classifier $g_{\boldsymbol{\varphi}^t}$ to make predictions. The parameter $\boldsymbol{\varphi}^t$ of the global classifier is then updated via gradient descent as:

$$\hat{\boldsymbol{\varphi}}^{t+1} \leftarrow \boldsymbol{\varphi}^t - \eta_{\boldsymbol{\varphi}} \nabla \mathcal{L}_{\varphi}(\boldsymbol{\varphi}^t; \{(\boldsymbol{\mu}_{c,k}^t, c)\}). \quad (8)$$

$\eta_{\boldsymbol{\varphi}}$ is the learning rate. $\mathcal{L}_{\varphi}(\boldsymbol{\varphi}^t; \{(\boldsymbol{\mu}_{c,k}^t, c)\})$ is defined as:

$$\mathcal{L}_{\varphi}(\boldsymbol{\varphi}^t; \{(\boldsymbol{\mu}_{c,k}^t, c)\}) = \sum_{c\in[C_k]} \frac{1}{|C_k|} \mathcal{L}_{CE}(g_{\boldsymbol{\varphi}^t}(\boldsymbol{\mu}_{c,k}^t), c). \quad (9)$$

Here, C_k denotes the number of classes on client k. The loss function $\mathcal{L}_{\varphi}(\boldsymbol{\varphi}^t; \{(\boldsymbol{\mu}_{c,k}^t, c)\})$ is defined as the cross-entropy loss, calculated over all the classes available on client k. The global classifier leverages the class-wise representation means from different clients to update its parameters, enabling it to gain a comprehensive understanding of all classes, leading to enhanced generalization capabilities compared to local classifiers with limited class-specific knowledge.

To improve global classifier training effectiveness, we propose a mechanism where the server trains the global classifier based on the class-wise representation means and covariances from each client. Once all participating clients' class-wise representation means and covariances are used to train the global classifier, it is updated in the current communication round. This process allows the global classifier to effectively learn from diverse client data without directly accessing raw data, maintaining privacy and security during the FL process.

4.4 Global Classifier Calibration

To thoroughly train the global classifier and leverage the uploaded class-wise representation mean and covariance $\{(\boldsymbol{\mu}_{c,k}^t, \boldsymbol{\Sigma}_{c,k}^t)\}_{c\in[C]}$ from each client k, we incorporate a mechanism that utilizes virtual samples generated from an approximated Gaussian mixture model [Luo *et al.*, 2021].

After receiving the class-wise representation means and covariances from S^t, the updated global classifier calculates the global mean $\boldsymbol{\mu}_c^t$ and covariance $\boldsymbol{\Sigma}_c^t$ for each class c as:

$$\boldsymbol{\mu}_c^t = \frac{1}{\sum_{k\in S^t} |D_k^c|} \sum_{k\in S^t} |D_k^c| \boldsymbol{\mu}_{c,k}^t. \quad (10)$$

To calculate the global covariance $\boldsymbol{\Sigma}_c^t$, we first obtain $(|D_k^c|-1)\boldsymbol{\Sigma}_{c,k}^t = \sum_{(\boldsymbol{x}_i,y_i)\in D_k^c} \boldsymbol{z}_{k,i}^t (\boldsymbol{z}_{k,i}^t)^T - |D_k^c| \boldsymbol{\mu}_{c,k}^t (\boldsymbol{\mu}_{c,k}^t)^T$. Let $D^c = \sum_{k\in S^t} D_k^c$. Then,

$$\begin{aligned}
&\boldsymbol{\Sigma}_c^t \\
&= \frac{1}{|D^c|-1} \sum_{k\in S^t} \sum_{(\boldsymbol{x}_i,y_i)\in D_k^c} \boldsymbol{z}_{k,i}^t (\boldsymbol{z}_{k,i}^t)^T - \frac{|D^c|}{|D^c|-1} \boldsymbol{\mu}_{c,k}^t (\boldsymbol{\mu}_{c,k}^t)^T \\
&= \sum_{k\in S^t} \frac{|D_k^c|}{|D^c|-1} \boldsymbol{\Sigma}_{c,k}^t + \sum_{k\in S^t} \frac{|D_k^c|}{|D^c|-1} \boldsymbol{\mu}_{c,k}^t (\boldsymbol{\mu}_{c,k}^t)^T \\
&\quad - \frac{|D^c|}{|D^c|-1} \boldsymbol{\mu}_c^t (\boldsymbol{\mu}_c^t)^T.
\end{aligned} \quad (11)$$

With the calculated global mean $\boldsymbol{\mu}_c^t$ and global covariance $\boldsymbol{\Sigma}_c^t$, we produce a collection G_c of virtual features associated with the true class c using the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_c^t, \boldsymbol{\Sigma}_c^t)$. Such production of virtual features allows us to model the distribution of data more comprehensively. To ensure that the virtual features reflect the inter-class distribution, the count of virtual features allocated to each class c, denoted as $|G_c|$, is determined based on the fraction $\frac{D^c}{\sum_{c\in[C]} D^c}$.

Algorithm 1 DC-PFL

INPUT: The number of rounds T; the total number of clients K; the number of clients selected in each training round S; the learning rate η for local models; the learning rate η_{φ} for the global classifier; the number of selected virtual features $|G_c|$ for class c; the hyperparameter λ balancing the supervised loss and auxiliary loss
Randomly initialize $\{\boldsymbol{w}_k^0\}_{k\in[K]}$ and $\boldsymbol{\varphi}^0$

ServerExecute

for $t = 2$ to T **do**
  $S^t \leftarrow$ the set of S randomly selected clients
  for client $k \in S^t$ **do**
    $\{(\boldsymbol{\mu}_{c,k}^t, \boldsymbol{\Sigma}_{c,k}^t)\}_{c\in[C]} \leftarrow$ ClientExecute(k, $\{\boldsymbol{\mu}_c^t\}_{c\in[C]}$, $\boldsymbol{\varphi}^t$)
    Update $\boldsymbol{\varphi}^t$ to $\hat{\boldsymbol{\varphi}}^{t+1}$ according to Eq. (8)
  end for
  Calculate the global mean $\boldsymbol{\mu}_c^t$ and covariance $\boldsymbol{\Sigma}_c^t$ for each class $c \in [C]$ according to Eq. (10) and (11)
  Draw virtual representations G_c for each class $c \in [C]$
  Get $\boldsymbol{\varphi}^{t+1}$ based on loss defined in Eq. (12)
end for

ClientExecute(k, $\{\boldsymbol{\mu}_c^t\}_{c\in[C]}$, $\boldsymbol{\varphi}^t$)

Updates the local classification model to $\hat{\boldsymbol{w}}_k^t = (\hat{\boldsymbol{\theta}}_k^t, \hat{\boldsymbol{\varphi}}^t)$
Calculate the total local loss according to Eq. (4)
Update $\hat{\boldsymbol{w}}_k^t$ to $\boldsymbol{w}_k^t$ according to Eq. (5)
Calculate the local mean $\boldsymbol{\mu}_{c,k}^t$ and covariance $\boldsymbol{\Sigma}_{c,k}^t$ for each class $c \in [C]$ according to Eq. (6) and (7)
Return $\{(\boldsymbol{\mu}_{c,k}^t, \boldsymbol{\Sigma}_{c,k}^t)\}_{c\in[C]}$

DC-PFL then fine-tunes the global classifier using the virtual representations G_c and their corresponding labels c. Specifically, the parameter of the global classifier $\hat{\boldsymbol{\varphi}}^{t+1}$, obtained in Eq. (8), is retrained to $\boldsymbol{\varphi}^{t+1}$ with the objective of minimizing the following calibration loss:

$$\mathcal{L}_{cali}(\hat{\boldsymbol{\varphi}}^{t+1}; \{G_c\}_{c\in[C]}) = \frac{1}{\sum_{c\in[C]} |G_c|} \sum_{(\boldsymbol{z},y)\in G_c} \mathcal{L}_{CE}(g_{\hat{\boldsymbol{\varphi}}^{t+1}}(\boldsymbol{z}), y). \tag{12}$$

After obtaining $\boldsymbol{\varphi}^{t+1}$ for the global classifier, the central server triggers training round $t+1$, broadcasting $\boldsymbol{\varphi}^{t+1}$ as well as $\boldsymbol{\mu}_c^{t+1} \leftarrow \boldsymbol{\mu}_c^t$ for each class c to the selected clients S^{t+1}.

These above four steps are iterated until all local heterogeneous models converge. In the first round, as the central server has not received any local mean and local covariance representations from the clients, it only broadcasts the global classifier $\boldsymbol{\varphi}^t$. Additionally, in the first round, the selected clients only calculate the supervised loss defined in Eq. (2) and update their local parameters based on the supervised loss alone. Algorithm 1 summarizes the training procedure for the proposed DC-PFL. This approach allows the FL server to effectively leverage knowledge from all clients, while preserving data privacy by only sharing class-wise representations.

5 Convergence Analysis

Consider each communication round t consisting of E local training epochs, assuming that the loss function is minimized through SGD. Let e represent the local epoch, where $e \in \{\frac{1}{2}, 1, \ldots, E\}$. In this context, $e = \frac{1}{2}$ corresponds to the epoch between the conclusion of FL server aggregation in round $(t-1)$, and the initiation of the first epoch of local training in round t. Upon completing E epochs of local training in round t, FL client k's local model can be denoted as $(\boldsymbol{\theta}_k^{E,t}, \boldsymbol{\varphi}^{E,t})$. Moving on to communication round $(t+1)$, k initializes the local model with the aggregated global model, denoted as $(\boldsymbol{\theta}_k^{\frac{1}{2},t+1}, \boldsymbol{\varphi}^{\frac{1}{2},t+1})$.

Assumption 5.1. (Lipschitz Continuity). We assume that the gradient of the local loss function $\mathcal{L}(\cdot)$ exhibits L_1-Lipschitz continuity, and the embedding functions of the local feature extractor $f_{\boldsymbol{\theta}}$ exhibits L_2-Lipschitz continuity:

$$\left\|\nabla\mathcal{L}(\boldsymbol{w}^{t_1}) - \nabla\mathcal{L}(\boldsymbol{w}^{t_2})\right\|_2 \le L_1 \left\|\boldsymbol{w}^{t_1} - \boldsymbol{w}^{t_2}\right\|_2, \forall t_1, t_2 > 0, \tag{13}$$

$$\left\|f_{\boldsymbol{\theta}^{t_1}} - f_{\boldsymbol{\theta}^{t_2}}\right\| \le L_2 \left\|\boldsymbol{\theta}^{t_1} - \boldsymbol{\theta}^{t_2}\right\|_2, \quad \forall t_1, t_2 > 0, \tag{14}$$

Assumption 5.2. (Unbiased Gradient and Bounded Variance). We assume the stochastic gradients $\boldsymbol{g}_k^t = \nabla\mathcal{L}\left(\boldsymbol{w}_k^t, \xi_k^t\right)$ computed on a batch of data ξ_k of client k are unbiased estimators of k' local gradient, which means:

$$\mathbb{E}_{\xi_k\sim D_k}\left[\boldsymbol{g}_k^t\right] = \nabla\mathcal{L}\left(\boldsymbol{w}_k^t\right) \quad \forall k \in [K], \tag{15}$$

with the variance bounded by σ^2,

$$\mathbb{E}\left[\left\|\boldsymbol{g}_k^t - \nabla\mathcal{L}\left(\boldsymbol{w}_k^t\right)\right\|_2^2\right] \le \sigma^2, \quad \forall k \in [K],\ \sigma > 0. \tag{16}$$

Assumption 5.3. (Gradients' Bounded Expectation). We assume that the expectation of the stochastic gradient is confined within the bound V:

$$\mathbb{E}\left[\left\|\boldsymbol{g}_k^t\right\|_2^2\right] \le V^2, \quad \forall k \in [K],\ V > 0. \tag{17}$$

Based on the above assumptions, we could get:

Lemma 5.4. *After E local training epochs, the loss function in communication round $(t+1)$ is bounded by:*

$$\mathbb{E}\left[\mathcal{L}^{E,t+1}\right] \le \mathcal{L}^{\frac{1}{2},t+1} - \sum_{e=\frac{1}{2}}^{E-1}\left(\eta_e - \frac{\eta_e^2 L_1}{2}\right)\left\|\nabla\mathcal{L}^{e,t+1}\right\|_2^2 + \frac{\eta_0^2 L_1 E}{2}\sigma^2. \tag{18}$$

Here, η_e denotes the learning rate at local epoch e.

Lemma 5.5. *The loss function of each client k at communication round $(t+1)$ after aggregating the model and mean class representations at the server is bounded by:*

$$\mathbb{E}\left[\mathcal{L}_k^{\frac{1}{2},(t+1)}\right] \le \mathcal{L}_k^{E,t} + \frac{\eta_0^2 L_1}{2}E^2V^2 + 2\lambda\eta_0 L_2 EV. \tag{19}$$

Theorem 5.6. *After communication round t, the loss function of each client k is bounded by:*

$$\mathbb{E}\left[\mathcal{L}_k^{\frac{1}{2},t+1}\right] \le \mathcal{L}_k^{\frac{1}{2},t} - \sum_{e=\frac{1}{2}}^{E-1}\left(\eta_e - \frac{\eta_e^2 L_1}{2}\right)\left\|\nabla\mathcal{L}^{e,t}\right\|_2^2 + \frac{\eta_0^2 L_1 E}{2}\left(EV^2 + 2\lambda\eta_0 L_2 EV + \sigma^2\right). \tag{20}$$

Theorem 5.7. *(Convergence of DC-PFL). If $\eta_0 > \eta_e > \alpha\eta_0$ for $e \in [1, E-1], 0 < \alpha < 1$, client k's loss function monotonically decreases in communication round t when*

$$\alpha\eta_0 < \eta_e < \frac{2\alpha^2\left\|\nabla\mathcal{L}^{e,t}\right\| - 4\alpha\lambda L_2 V}{\left(EV^2+\sigma^2\right)L_1\left(\alpha^2\left\|\nabla\mathcal{L}^{e,t}\right\|_2^2 + 1\right)}, \forall e \in [1, E-1]. \tag{21}$$

α is the hyper-parameter controlling learning rate decay.

Theorem 5.8. *(Convergence Rate of DC-PFL). Define regret $\Delta = \mathcal{L}^{\frac{1}{2},1} - \mathcal{L}^*$ and incorporate Assumptions 1-3. After $T = \frac{2\Delta}{\epsilon E(2\eta - \eta^2 L_1) - \eta^2 L_1 E(EV^2+\sigma^2) - 4\lambda\eta L_2 EV}$ communication rounds with $\epsilon > 0$ and learning rate η,*

$$\frac{1}{TE}\sum_{t=1}^{T}\sum_{e=\frac{1}{2}}^{E-1}\left\|\nabla\mathcal{L}^{e,t}\right\|_2^2 \le \epsilon \tag{22}$$

holds for each client k.

Detailed proofs of the above lemmas, theorems and corollary can be found in Appendix A.

6 Experimental Evaluation

6.1 Experiment Settings

Datasets. We assess the performance of the proposed DC-PFL alongside baselines on datasets CIFAR-10 and CIFAR-100[1]. To create non-IID versions of these datasets, we follow the method outlined in [Yi *et al.*, 2023]. Specifically, in CIFAR-10, each client is allocated data from only 2 classes (non-IID: 2/10), while in CIFAR-100, each client receives data from only 10 classes (non-IID: 10/100).

Furthermore, the data from each client is partitioned into three distinct subsets: training, evaluation, and testing, with

[1]https://www.cs.toronto.edu/ kriz/cifar.html

an 8:1:1 allocation ratio. Notably, each client retains the testing set locally, ensuring that it reflects the distribution of their specific local training set. The CNN models used are identical to those outlined in [Yi *et al.*, 2023]. In these models, the dimensions of the representation layer (i.e., the second-to-last layer) are consistently set at 500 while the classification layer (i.e., the last fully-connected layer) is configured as 10 and 100, respectively.

Comparison Baselines. We experimentally compare DC−PFL with the following eight baseline methods. *Local-T*, each client independently trains its local model; *FedAvg* [McMahan *et al.*, 2017], a widely recognized FL that exclusively supports homogeneous local models; *FML* [Shen *et al.*, 2020] and *FedKD* [Wu *et al.*, 2022], the mutual learning-based methods; *LG-FedAvg* [Liang *et al.*, 2020] and *FedGH* [Yi *et al.*, 2023], the model mixup-based PFL method; *FD* [Jeong *et al.*, 2018] and *FedProto* [Tan *et al.*, 2022b], the knowledge distillation-based methods.

We optimize FL hyperparameters through an extensive grid search by adjusting the batch size for local training from {32, 64, 128, 256, 512} and the number of local training epochs from {1, 10, 30, 50, 100}. We utilize the SGD optimizer with a fixed learning rate (η) of 0.01 for both local training and global classifier training. The total number of communication rounds (T) is set to 100 on CIFAR-10 and to 500 on CIFAR-100 to ensure convergence across all algorithms.

Method	$\|K\|=10$, $\frac{\|S^t\|}{\|K\|}=100\%$		$\|K\|=50$, $\frac{\|S^t\|}{\|K\|}=20\%$		$\|K\|=100$, $\frac{\|S^t\|}{\|K\|}=10\%$	
	CIFAR10	CIFAR100	CIFAR10	CIFAR100	CIFAR10	CIFAR100
Local-T	92.57	59.62	94.74	59.35	91.63	53.19
FedAvg	93.78	61.33	94.98	59.14	92.08	53.36
FML	92.01	59.04	93.95	53.68	89.01	47.74
FedKD	92.84	55.38	93.47	54.47	89.87	49.57
LG-FedAvg	93.02	61.07	94.72	58.25	91.52	52.98
FD	93.11	-	-	-	-	-
FedProto	95.42	59.86	94.83	58.63	91.34	53.42
FedGH	95.96	72.00	95.01	59.88	92.15	53.93
DC−PFL	**96.31**	**74.93**	**95.93**	**67.45**	**93.18**	**60.20**
w/o AL	96.08	74.43	95.51	66.10	92.56	60.05
w/o GC	95.97	72.56	95.76	61.66	92.32	54.61

Table 1: Test accuracy (%) comparison in the model homogeneous FL settings with varied number of clients $|K|$ and client participation rates $\frac{|S^t|}{|K|}$. "-" means that the corresponding algorithm does not achieve convergence.

Method	CIFAR10 (2/10)	CIFAR100 (10/100)
Local-T	92.84	70.84
FML	-	-
FedKD	78.05	53.45
LG-FedAvg	93.95	70.73
FD	94.47	-
FedProto	94.36	71.01
FedGH	96.08	71.56
DC−PFL	**96.16**	**74.70**

Table 2: Test accuracy (%) comparison in the model heterogeneous FL settings. "-" means that the corresponding algorithm does not achieve convergence.

6.2 Results and Discussion

To conduct a comprehensive comparison between the proposed DC−PFL and existing methods, we begin by evaluating their performance in model homogeneity FL settings and subsequently transition to model heterogeneity FL settings.

Model Homogeneity FL setting. In line with [Yi *et al.*, 2023], our experiments encompass three distinct scenarios, each characterized by varying numbers of clients, denoted as $|K|$, and client participation rates $\frac{|S^t|}{|K|}$: 1) $|K|=10$, $\frac{|S^t|}{|K|}=100\%$. 2) $|K|=50$, $\frac{|S^t|}{|K|}=20\%$. 3) $|K|=100$, $\frac{|S^t|}{|K|}=10\%$. In each case, the number of clients participating in the FL training process per round is held constant to ensure fairness and enable a comprehensive comparison across diverse settings. Specifically, the number of participating clients per round is fixed at $|K| \times \frac{|S^t|}{|K|} = 10$. The results are shown in Table 1.

Our observations reveal that DC−PFL consistently outperforms state-of-the-art methods, including those specifically designed for model homogeneity FL settings (such as FedAvg) and methods tailored for model heterogeneity FL settings, encompassing techniques ranging from mutual learning (FML, FedKD) and model mixup (LG-FedAvg, FedGH) to knowledge distillation on representations within the same class (FedProto, FedGH) and knowledge distillation on logits within the same class (FD). Compared with the best baseline FedGH, DC−PFL improves the accuracy by 1.22% and 9.22% on average for CIFAR-10 and CIFAR-100, respectively. This improvement is particularly remarkable considering that most algorithms already achieve high accuracy on CIFAR-10. Furthermore, the substantial accuracy enhancement of DC−PFL on CIFAR-100 underscores its efficacy in addressing the non-IID issues (i.e., statistical heterogeneity).

Model Heterogeneity FL setting. To comprehensively compare DC−PFL with existing methods under model heterogeneity FL settings, we introduce variability by adjusting the dimensions of the fully-connected layers and the number of filters within the convolutional layers of our CNN model. This approach follows the methodology in [Yi *et al.*, 2023]. We then evenly distribute these models among the clients, with the possibility of different clients possessing models of identical structures. The results are summarized in Table 2.

It can be observed that DC−PFL outperforms existing methods in terms of model accuracy within the model heterogeneity FL settings. Notably, compared to the best-performing baseline, FedGH, DC−PFL exhibits average accuracy improvements of 4.38% for CIFAR-100. These results unequivocally underscore the effectiveness of DC−PFL.

Ablation study. We devised two ablated versions of DC−PFL to gain deeper insights into its components: 1) "w/o AL": In this ablated version, we exclude the auxiliary loss from the local training loss. 2) "w/o GC": This ablated version omits the global classifier-level calibration (i.e., the Gaussian generation process). The results of the ablation experiments are presented in Table 1. Notably, DC−PFL outperforms its ablated variants, highlighting the effectiveness of the dual calibration design in enhancing model mixup-based PFL performance. Compared to "w/o GC", "w/o AL" performs better in most cases. It may indicate that the effectiveness of DC−PFL is largely due to the incorporation of the global classifier-level calibration.

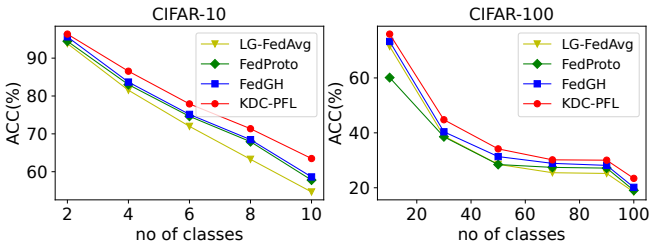

Figure 2: Comparison under different numbers of classes.

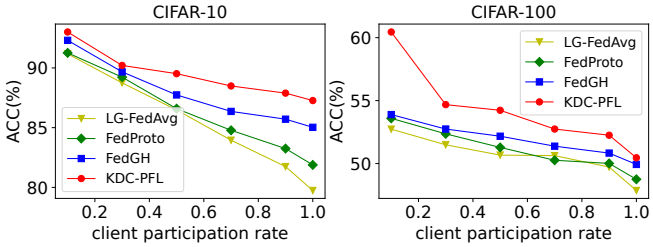

Figure 3: Comparison under varying client participation rates.

Robustness to non-IIDness. We compare DC−PFL with existing model-heterogeneous baselines, namely FedProto, LG-FedAvg, and FedGH, on both datasets, incorporating varying degrees of non-IID data distributions. Specifically, we set the number of clients $|K|$ to 10, and client participation rates $\frac{|S^t|}{|K|}$ to 100%. For the CIFAR-10 dataset, each client is distributed with samples from $\{2, 4, 6, 8, 10\}$ classes, while for CIFAR-100, each client is allocated samples from $\{10, 30, 50, 70, 90, 100\}$ classes, with a higher number of classes indicating a lower degree of non-IIDness. Figure 2 clearly illustrates that DC−PFL consistently achieves the highest model accuracy across various degrees of non-IID data distribution on both CIFAR-10 and CIFAR-100 datasets. This outstanding performance demonstrates its robustness in handling non-IID data. Moreover, with an increase in the number of classes (indicating a more IID dataset), the model accuracy exhibits a decreasing trend. This aligns with the observation that the benefits of personalizing local models diminish as data heterogeneity decreases, corroborating with [Shen *et al.*, 2020].

Robustness to participation rate. We compare DC−PFL with existing model-heterogeneous baselines, FedProto, LG-FedAvg and FedGH on both datasets with varying participation rates of clients. Specifically, we fix the number of clients at $|K| = 100$ and vary the client participation rates $\frac{|S^t|}{|K|} \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ for CIFAR-10 (non-IID: 2/10) and CIFAR-100 (non-IID: 10/100). Figure 3 shows that DC−PFL consistently outperforms existing methods across all client participation rate settings on both CIFAR-10 and CIFAR-100. This underscores the robustness of DC−PFL in handling varying client participation rates. Notably, the model accuracy tends to decrease as the client participation rate increases. This observation can be attributed to the fact that as more clients engage in a communication round, there is an improvement in generalization, while the task of personalization becomes more challenging.

Method	$\|K\| = 10$, $\frac{\|S^t\|}{\|K\|} = 100\%$	$\|K\| = 50$, $\frac{\|S^t\|}{\|K\|} = 20\%$	$\|K\| = 100$ $\frac{\|S^t\|}{\|K\|} = 10\%$
FedGH	72.00	59.88	63.93
DC−PFL ($\sum \|G_c\| = 600$)	74.33	66.93	58.78
DC−PFL ($\sum \|G_c\| = 800$)	74.56	67.23	60.27
DC−PFL ($\sum \|G_c\| = 1,000$)	74.93	67.45	60.20
DC−PFL ($\sum \|G_c\| = 2,000$)	74.34	67.67	59.27
DC−PFL ($\sum \|G_c\| = 5,000$)	74.50	67.38	59.94

Table 3: Accuracy (%) of the FL models produced by DC−PFL with different numbers of generated virtual samples on CIFAR-100.

Sensitivity to the number of generated virtual samples. To assess the impact of the number of generated virtual samples $\sum |G_c|$, we vary the number of generated virtual samples among $\{600, 800, 1,000, 2,000, 5,000\}$ on the CIFAR-100 dataset. The results are presented in Table 3. The results indicate an interesting trend concerning the accuracy with respect to the number of generated virtual samples. Initially, as this number increases, there is a notable rise in accuracy, followed by a subsequent decline. This pattern suggests that a higher number of generated virtual samples initially enriches the ability to mimic the true feature distribution, thereby enhancing overall performance. However, as the number of generated virtual samples continues to grow, there is a risk of deviating from the true feature distribution, leading to a drop in performance. Table 3 shows that selecting a total number of generated virtual samples between 1,000 to 2,000 yields optimal results for DC−PFL. Therefore, we set $\sum |G_c| = 1,000$.

Visualizing Personalization. We extracted representations for each sample from each FL client in DC−PFL and FedGH. We then used T-SNE to reduce the dimensionality of the representations from 500 to 2 for visualization. More details and results are in Appendix B.

7 Conclusions

In this paper, we propose a novel method, DC−PFL, to enhance model mixup-based PFL. Under DC−PFL, client local models consist of a heterogeneous feature extractor and a homogeneous classifier. The FL server utilizes mean and covariance representations from clients' feature extractors to train a global generalized classifier to be shared among all clients, facilitating information exchange while preserving privacy. An auxiliary loss guides local models to improve feature extractors by using global knowledge. The global classifier is fine-tuned with sample representations derived from an approximate Gaussian distribution model specific to each class. DC−PFL eliminates the need to transmit original data representations, thus enhancing privacy preservation. It enable FL clients to build personalized heterogeneous local models tailored to their unique data distributions, system resources, and model structures. The FL server fosters collaboration and knowledge sharing, leading to improved model performance. Notably, the design of virtual representations and knowledge distillation ensures robust global model training while addressing privacy concerns.

Acknowledgments

This research is supported, in part, by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (No. AISG2-RP-2020-019); and RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore.

References

[Chang *et al.*, 2019] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv preprint arXiv:1912.11279*, 2019.

[Chen *et al.*, 2021] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. Fedmatch: federated learning over heterogeneous question answering data. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM'21)*, pages 181–190, 2021.

[Cheng *et al.*, 2021] Sijie Cheng, Jingwen Wu, Yanghua Xiao, and Yang Liu. Fedgems: Federated learning of larger server models via selective knowledge fusion. *arXiv preprint arXiv:2110.11027*, 2021.

[Cho *et al.*, 2022] Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI'22)*, pages 1881–1887, 2022.

[Collins *et al.*, 2021] 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 (ICML'21)*, pages 2089–2099, 2021.

[Huang *et al.*, 2022a] Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, pages 10143–10153, 2022.

[Huang *et al.*, 2022b] Wenke Huang, Mang Ye, Bo Du, and Xiang Gao. Few-shot model agnostic federated learning. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM'22)*, pages 7309–7316, 2022.

[Itahara *et al.*, 2021] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing*, 22(1):191–205, 2021.

[Jang *et al.*, 2022] Jaehee Jang, Heoneok Ha, Dahuin Jung, and Sungroh Yoon. Fedclassavg: Local representation learning for personalized federated learning on heterogeneous neural networks. In *Proceedings of the 51st International Conference on Parallel Processing (ICPP'22)*, pages 1–10, 2022.

[Jeong *et al.*, 2018] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[Li and Wang, 2019] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[Li *et al.*, 2021] Qinbin Li, Bingsheng He, and Dawn Song. Practical one-shot federated learning for cross-silo setting. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21)*, pages 1484–1490, 2021.

[Liang *et al.*, 2020] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

[Lin *et al.*, 2020] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS'20)*, pages 2351–2363, 2020.

[Liu *et al.*, 2022] Chang Liu, Yuwen Yang, Xun Cai, Yue Ding, and Hongtao Lu. Completely heterogeneous federated learning. *arXiv preprint arXiv:2210.15865*, 2022.

[Liu *et al.*, 2024] Rui Liu, Pengwei Xing, Zichao Deng, Anran Li, Cuntai Guan, and Han Yu. Federated graph neural networks: Overview, techniques, and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Luo *et al.*, 2021] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021.

[Makhija *et al.*, 2022] Disha Makhija, Xing Han, Nhat Ho, and Joydeep Ghosh. Architecture agnostic federated learning for neural networks. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, pages 14860–14870, 2022.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS'17)*, pages 1273–1282, 2017.

[Oh *et al.*, 2022] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*, 2022.

[Pillutla *et al.*, 2022] Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, pages 17716–17758, 2022.

[Robbins and Monro, 1951] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[Sattler *et al.*, 2021a] Felix Sattler, Tim Korjakow, Roman Rischke, and Wojciech Samek. Fedaux: Leveraging unlabeled auxiliary data in federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[Sattler *et al.*, 2021b] Felix Sattler, Arturo Marban, Roman Rischke, and Wojciech Samek. Cfd: Communication-efficient federated distillation via soft-label quantization and delta coding. *IEEE Transactions on Network Science and Engineering*, 9(4):2025–2038, 2021.

[Shen *et al.*, 2020] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Kun Kuang, Fei Wu, and Chao Wu. Federated mutual learning. *arXiv preprint arXiv:2006.16765*, 2020.

[Tan *et al.*, 2022a] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Tan *et al.*, 2022b] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022.

[Tan *et al.*, 2024] Yue Tan, Chen Chen, Weiming Zhuang, Xin Dong, Lingjuan Lyu, and Guodong Long. Is heterogeneity notorious? taming heterogeneity to handle test-time shift in federated learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[Wu *et al.*, 2022] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1):2032, 2022.

[Yi *et al.*, 2023] Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. FedGH: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM International Conference on Multimedia (ACM MM'23)*, 2023.

[Yu *et al.*, 2022] Sixing Yu, Wei Qian, and Ali Jannesari. Resource-aware federated learning using knowledge extraction and multi-model fusion. *arXiv preprint arXiv:2208.07978*, 2022.

[Zhang *et al.*, 2022] Lan Zhang, Dapeng Wu, and Xiaoyong Yuan. Fedzkt: Zero-shot knowledge transfer towards resource-constrained federated learning with heterogeneous on-device models. In *Proceedings of the IEEE 42nd International Conference on Distributed Computing Systems (ICDCS'22)*, pages 928–938. IEEE, 2022.

[Zhu *et al.*, 2021] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, pages 12878–12889, 2021.

[Zhuang *et al.*, 2023] Weiming Zhuang, Chen Chen, and Lingjuan Lyu. When foundation model meets federated learning: Motivations, challenges, and future directions. *arXiv preprint arXiv:2306.15546*, 2023.