FedCD: A Classifier Debiased Federated Learning Framework for Non-IID Data

Yunfei Long
Beijing University of Posts and Telecommunications
Beijing, China
longyunfei@bupt.edu.cn

Zhe Xue*
Beijing University of Posts and Telecommunications
Beijing, China
xuezhe@bupt.edu.cn

Lingyang Chu
McMaster University
Hamilton, Canada
chul9@mcmaster.ca

Tianlong Zhang
Beijing University of Posts and Telecommunications
Beijing, China
tlzhang@bupt.edu.cn

Junjiang Wu
Beijing University of Posts and Telecommunications
Beijing, China
wujunjiang@bupt.edu.cn

Yu Zang
Beijing University of Posts and Telecommunications
Beijing, China
zyzy@bupt.edu.cn

Junping Du
Beijing University of Posts and Telecommunications
Beijing, China
junpingd@bupt.edu.cn

ABSTRACT

One big challenge to federated learning is the non-IID data distribution caused by imbalanced classes. Existing federated learning approaches tend to bias towards classes containing a larger number of samples during local updates, which causes unwanted drift in the local classifiers. To address this issue, we propose a **c**lassifier **d**ebiased federated learning framework named FedCD for non-IID data. We introduce a novel hierarchical prototype contrastive learning strategy to learn fine-grained prototypes for each class. The prototypes characterize the sample distribution within each class, which helps align the features learned in the representation layer of every client's local model. At the representation layer, we use fine-grained prototypes to rebalance the class distribution on each client and rectify the classification layer of each local model. To alleviate the bias of the classification layer of the local models, we incorporate a global information distillation method to enable the local classifier to learn decoupled global classification information. We also adaptively aggregate the class-level classifiers based on their quality to reduce the impact of unreliable classes in each aggregated classifier. This mitigates the impact of client-side classifier bias on the global classifier. Comprehensive experiments conducted on various datasets show that our method, FedCD, effectively corrects classifier bias and outperforms state-of-the-art federated learning methods.

CCS CONCEPTS

• **Security and privacy** → **Domain-specific security and privacy architectures**; • **Computing methodologies** → *Machine learning algorithms.*

KEYWORDS

federated learning, prototype learning, knowledge distillation

ACM Reference Format:
Yunfei Long, Zhe Xue, Lingyang Chu, Tianlong Zhang, Junjiang Wu, Yu Zang, and Junping Du. 2023. FedCD: A Classifier Debiased Federated Learning Framework for Non-IID Data. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3611966

1 INTRODUCTION

Real-world data is usually distributed across multiple clients, such as mobile devices, hospitals and institutions. Due to privacy and data protection concerns [41], data on each client can only be accessed locally. Instead of traditional centralized learning methods, federated learning (FL) [6, 30, 33, 44, 45] has been proposed to address the data silos problem and protect users' data privacy without information leakage. The FL process consists of local training on clients and global model aggregation on the server. It has been successfully applied in numerous real-world applications, including the Internet of Things [13, 38], health care [11, 12] and multimedia analysis [16, 27, 32]

Traditional federated learning methods, such as FedAvg [33], have demonstrated significant success in scenarios where data is independent and identically distributed (IID). However, in real-world

*Corresponding author.

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3611966

scenarios, data samples across different clients often exhibit non-IID characteristics. One of the key challenges in non-IID federated learning is the presence of class imbalance. The existence of imbalanced data can cause FedAvg to favor classes with a larger number of samples during local updates, leading to client drift. Moreover, research has shown that the classification layer in local deep models introduces more bias than the hidden layers in non-IID federated learning [31], which significantly deviates from the global optimum in terms of local optimization goals. As a result, the performance of FedAvg deteriorates considerably under these circumstances. This issue is also an urgent problem to be addressed in non-IID federated learning.

Many federated learning approaches have been proposed to solve non-IID federated learning. Classic methods include Fedprox [29], Scaffold [23], FedDC [15], Lumos [36] and FedIR [21], which propose local optimization constraints to align local and global optimization goals. Other methods include FedNova [43], FedMA [42] , FedAvgM [20] and CCVR [31] to make the global model close to the global optimum by improving the global aggregation stage. Nevertheless, despite the advancements made by these methods, they still fall short in addressing the challenge of classifier bias that arises with imbalanced classes. In the following analysis, we will discuss the limitations of these methods to elucidate the underlying causes of this issue.

First, the primary factor contributing to this issue is the inability of clients to access the distribution information of global sample features during local updates, as Figure 1(a). As a result, the learned feature representations may lack highly separable characteristics. Second, during the local update process, the progressive loss of global classification information leads to a bias of the local classifier towards the local optimum. Third, during the global aggregation stage, the varying importance of knowledge learned by the same client for different classes is not considered, which may further cause low-quality classes in the client to negatively affect the aggregation process of the global model.

To address the aforementioned challenges, we propose FedCD, a framework that includes hierarchical prototype contrastive learning, global information distillation, and adaptive class-level classifier aggregation. To overcome the first limitation, we propose hierarchical prototype contrastive learning to learn fine-grained prototypes for each class, improving the characterization of their sample distribution and enhancing the hierarchical separability between fine-grained classes. By rebalancing the sample distribution using the fine-grained prototypes, clients gain access to global sample distribution information, as shown in Figure 1(b). To enhance feature separability within each batch, we introduce batch prototype regularization loss, which makes the learned representations fit into the input space of the global classifier. For the second limitation, we introduce global information distillation, which decouples the soft labels output by the global classifier. This approach can align the local classifier and the global classifier at the decision level, and alleviate the problem of local classifiers' bias towards a large number of classes in the client. To address the third limitation, we propose adaptive class-level classifier aggregation, which involves partitioning classifiers into fine-grained class-vectors and adaptively evaluating their quality. Higher weights are assigned to high-quality class-vectors, while lower weights are assigned to

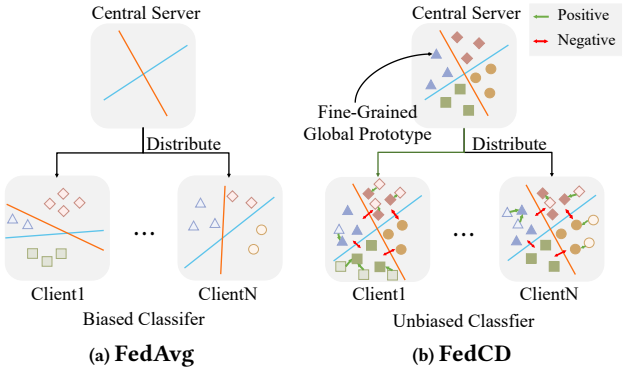


Figure 1: Comparison between FedAvg and FedCD. By learning the fine-grained global prototypes, FedCD can debias the classifier drift on client.

unreliable class-vectors. This approach effectively mitigates the influence of unreliable class-vectors from clients on the global classifier. We conduct extensive experiments on several datasets, and the experimental results showcase the advantages of our method compared to other approaches. The main contributions of this paper can be summarized as follows:

- We propose hierarchical prototype contrastive learning loss and batch prototype regularization loss to learn fine-grained prototypes that accurately characterize the global sample distribution and encourage feature representations within classes to aggregate while remaining distant from other classes. Compared with single prototype methods, the proposed fine-grained prototypes rebalance the feature distribution of samples on clients more accurately, effectively reducing the bias of local classifiers.
- We propose a global information distillation loss to align local and global classifiers at the decision level, mitigating the bias of local classifiers towards a large number of classes on the client side. By decoupling the soft labels output by the global classifier, local classifiers can be debiased by global classification information that clients cannot access.
- We introduce an adaptive class-level classifier aggregation method that partitions classifiers into fine-grained class-vectors. By adaptively assigning weights to these vectors, we effectively mitigate the impact of unreliable class-vectors on the global classifier and further enhance the robustness of the aggregated global classifier.

2 RELATED WORK

Limited by non-IID data in the real world, federated learning [33] does not perform as well as expected. Extensive approaches are working on exploring non-IID federated learning solutions in client selection [3, 14, 40], aggregation scheme [4, 9, 10] and personalized federated learning [2, 8, 22, 28]. Then, we focus on the techniques that most correlate with our work

Client Drift Alleviation. Some vanilla federated learning algorithms alleviate client drift by modifying the local optimization objectives, e.g., Fedprox [29] directly uses the ℓ_2-norm distance to mitigate client drift, Scaffold [23] uses variance reduction to correct client drift, FedDC [15] uses the learned local drift variables

to bridge the gap, i.e., perform consistency constraints at the parameter level. FedDyn [1] dynamically optimizes local objectives iteratively to achieve asymptotic consistency between local optima and stationary points of the global objective, FedMA [42] matches and averages weights to build a shared global model in a layer-wise manner, CCVR [31] samples pseudo-samples on the server and calibrates classifiers to address classifier drift.

Contrastive Learning in Federated Learning. Contrastive learning [5, 18] has already shown excellent prospects in unsupervised representation learning. FedX [17] combines contrastive learning with federated learning and uses global and local distillation methods to learn the vector representation of samples without supervision. In supervised areas, MOON [26] firstly applies contrastive learning at the model level to make the local model learn a representation that is closer to the global model and better than the local models of the previous round.

In recent years, prototype learning has been widely developed, where class prototypes are represented as the average feature vector of the class [34]. FedProto [39] suggests minimizing communication overhead by exchanging prototypes rather than gradients or model parameters between the client and server, FedProc [35] introduces the utilization of global prototypes on the server as a reference for refining client training during local updates. It employs a contrastive loss to encourage intra-class features to be closer while inter-class features to be farther apart.

3 METHODOLOGY

3.1 Overall Framework

The framework of the proposed method FedCD is shown in Figure 2. We introduce each module in detail. 1) **Fine-Grained Prototype Learning** aligns the sample representation on each client and optimizes each fine-grained local prototype to accurately reflect the global sample distribution by $\mathcal{L}_{hpc}$. It also constrains the batch prototype to fit into the input place of the global classifier by $\mathcal{L}_{bpr}$; 2) **Global Information Distillation** empowers the local classifier to assimilate global classification information, by leveraging $\mathcal{L}_{gid}$, thereby establishing alignment between the local and global classifiers. Furthermore, it enables the adjustment of a biased local classifier based on the fine-grained global prototype using $\mathcal{L}_{dcl}$; 3) **Fine-Grained Global Prototype Aggregation** aggregates the fine-grained local prototypes of each fine-grained class into a fine-grained global prototype using the sample proportion of local clients as weights; 4) **Adaptive Class-Level Classifier Aggregation** utilizes fine-grained global prototypes to adaptively assess the quality of class vectors on the client level, thereby enhancing the robustness of the global classifier. This is achieved by emphasizing high-quality class vectors and reducing the weight of low-quality ones during the aggregation. 5) **Feature Extractor Aggregation** takes the same approach as Fedavg [33] to aggregate local feature extractors from all clients to obtain the global feature extractor.

3.2 Fine-Grained Global Prototype Aggregation

In this study, all samples belong to C ground-truth classes. Each class is further partitioned into M fine-grained classes, with each fine-grained class contains a fine-grained prototype $\mathrm{P}_{c,m}$, where $c \in [1, C]$, $m \in [1, M]$. During the global aggregation stage, our objective is to characterize the global data distribution information by aggregating fine-grained global prototypes P^g for each class. To accomplish this, we adopt a weighted aggregation method on the server, aggregating the fine-grained local prototypes from various clients. The weight assigned for aggregation is determined by the ratio of the number of samples in the class to the total number of samples in that class across all clients. Each fine-grained local prototype is obtained by optimizing through gradient descent during the local update phase.

After the server receives all fine-grained local prototypes belonging to the m-th fine-grained class under the c-th ground-truth class, the fine-grained global prototype $\mathrm{P}^g_{c,m}$ is obtained as:

$$\mathrm{P}^g_{c,m} = \sum_{k=1}^{K} \frac{n^k_{c,m}}{n_{c,m}} \mathrm{P}^k_{c,m}, \tag{1}$$

where $\mathrm{P}^k_{c,m}$ is the fine-grained local prototype from client k. $n^k_{c,m}$ and $n_{c,m}$ represent the number of samples for the corresponding fine-grained class on the k-th client and the total number of samples for that class across all clients, respectively.

3.3 Fine-Grained Prototype Learning

This paper aims to address the issue of class imbalance in non-IID federated learning. The presence of class imbalance often leads to insufficient samples for minority classes. Such imbalance poses challenges in accurately characterizing the true data distribution and hinders the effective extraction of relevant information during local model training. Furthermore, in the presence of class imbalance, local classifiers tend to exhibit bias towards the majority class, resulting in a decline in the overall performance of the global model. In real data distributions, samples of the same class cannot be compactly clustered in a cluster, resulting in the learning of a single prototype being insufficient to effectively describe the sample distribution of each class.[25, 37, 48]. To overcome these challenges, we propose a novel approach that involves learning fine-grained prototypes for each class. These fine-grained prototypes accurately capture the sample distribution for each class on clients and are used to mitigate the bias of the local classifier.

Fine-Grained Class Assignment. After receiving the fine-grained global prototypes P^g from the server, we proceed with assigning samples to fine-grained classes on the client. For i-th sample $x^k_{i,c}$ from class c on the k-th client, its feature is expressed as $z^k_{i,c} = f_e(x^k_{i,c}|\theta_k)$, and its fine-grained class is obtained as follows:

$$\begin{aligned} s^k_m &= sim(z^k_{i,c}, \mathrm{P}^g_{c,m}), \\ \mathrm{S}^k_i &= [s^k_1, \ldots, s^k_m, \ldots, s^k_M], \\ t &= \arg\max_m (\mathrm{S}^k_i), \end{aligned} \tag{2}$$

where $sim(\cdot,\cdot)$ is the similarity function, and in this work, we use the cosine similarity function. s^k_m is the similarity of $z^k_{i,c}$ to the m-th fine-grained global prototype of the c-th class. $f_e(\cdot|\theta_k)$ represents the feature extractor with its parameter. Finally, t is the assigned class label that results in the maximum value for S^k_i, and the feature of the assigned sample is further denoted by $z^k_{i,c,t}$.

Hierarchical Prototype Contrastive Learning. In contrast to existing methods using the average of samples from a class as prototypes, our method initializes fine-grained local prototypes

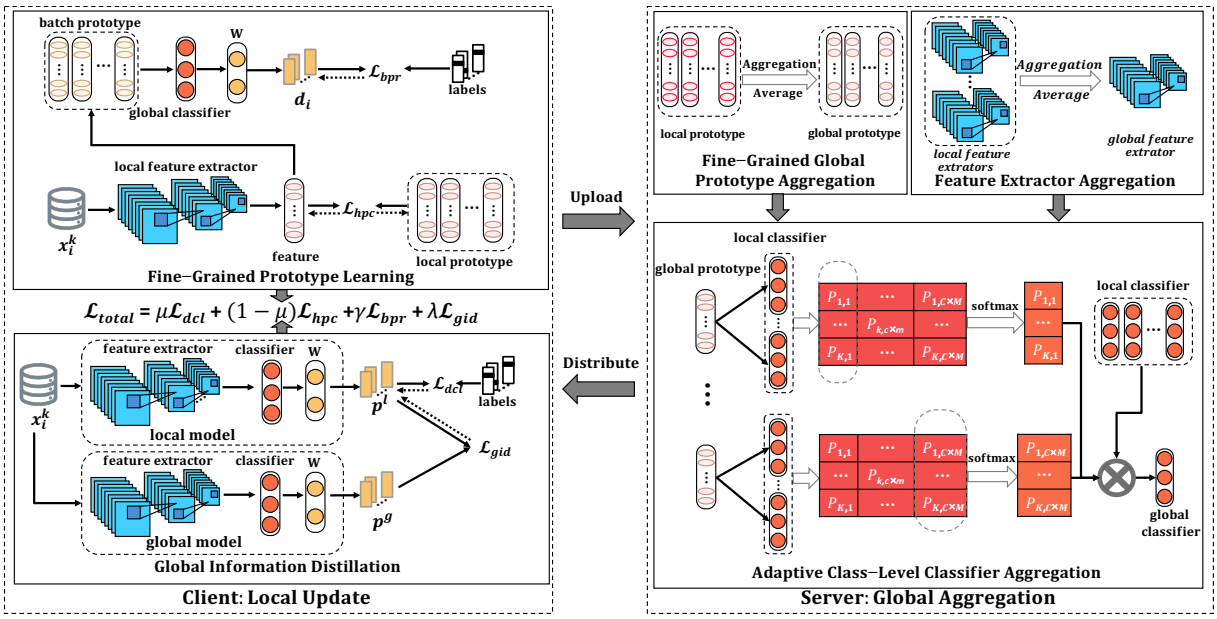


Figure 2: The framework of the proposed method FedCD. Solid arrows represent the forward propagation process, while dashed arrows indicate the direction of backpropagation process.

with fine-grained global prototypes (In the first round, we initialize the local prototype using the k-means) and treats them as learnable variables, optimized through gradient descent. Specifically, given a sample feature $z^k_{i,c,t}$, it should be closest to the local prototype of the fine-grained class it belongs to ($\mathrm{P}^k_{c,t}$), while maintaining an appropriate distance from other fine-grained local prototypes within the same class ($\mathrm{P}^k_{c,m}, m \neq t$), and exhibiting significant separation from local prototypes of different classes ($\mathrm{P}_{j,m}, j \neq c$). To achieve this, we propose a novel hierarchical prototype contrastive learning, constructing a positive pair Ψ_{pos}, $M-1$ pseudo negative pairs Ψ_{pneg}, and $(C-1)\times M$ negative pairs Ψ_{neg} to align sample representations and facilitate the learning of accurate fine-grained prototypes that capture the sample distribution. The loss is defined as follows:

$$\begin{aligned}\Psi_{pos} &= \psi(z^k_{i,c,t}, \mathrm{P}^k_{c,t}),\\ \Psi_{pneg} &= \sum_{m\neq t} \psi(z^k_{i,c,t}, \mathrm{P}^k_{c,m}),\\ \Psi_{neg} &= \sum_{j\neq c}\sum_{m} \psi(z^k_{i,c,t}, \mathrm{P}^k_{j,m}),\end{aligned} \tag{3}$$

where $\psi(z, \mathrm{P}) = e^{sim(z,\mathrm{P})/\tau}$, τ is the temperature parameter. We introduce a parameter α to adjust the hierarchical distance of features to two types of negative pairs. The loss of hierarchical prototype contrastive learning is defined as follows:

$$\omega(z^k_{i,c,t}) = \frac{\Psi_{pos}}{\Psi_{pos} + \alpha\Psi_{pneg} + (1-\alpha)\Psi_{neg}}, \tag{4}$$

$$\mathcal{L}_{hpc} = -\frac{1}{N}\sum_{z^k_{i,c,t}\in\mathcal{D}^{k+}} \mathrm{I}_{c,t} log\omega(z^k_{i,c,t}), \tag{5}$$

where $\mathcal{D}^{k+} = \{\mathcal{D}^k \cup \mathrm{P}^g\}$ is the rebalanced local dataset. $\mathcal{D}^k$ is the local sample features set and P^g is the set of fine-grained global prototypes. The importance factor $\mathrm{I}_{c,t}$ represents the significance of each sample. Local samples are assigned a value of 1 for $\mathrm{I}_{c,t}$. For the fine-grained global prototype $\mathrm{P}^g_{c,t}$, $\mathrm{I}_{c,t}$ is equal to $n_{c,t}$. $N = \sum_{c=1}^{C}\sum_{m=1}^{M} \mathrm{I}_{c,t}$ is the sum of importance factors. We set $\alpha < 0.5$ to indicate that the feature is farther from the real negative samples compared to the pseudo negative samples.

Batch Prototype Regularization. Enhancing the separability of feature representations is of paramount importance for classifier correction. In light of this, we introduce a batch prototype regularization to effectively align features with the input space of a global classifier. By incorporating this regularization loss, we aim to bolster the separability of feature representations, thereby enhancing the overall discriminative power:

$$d_i = \sigma(D(\mathrm{P}^b_c|\phi)\cdot W), 1 \leq i \leq N_b, c \in \mathcal{Y}_b, \tag{6}$$

where D is the global classifier and ϕ is its parameters. $W \in \mathbb{R}^{(C\times M)\times C}$ is an aggregation matrix employed to consolidate the classification results output by the fine-grained classifier. $\sigma(\cdot)$ is the softmax function. P^b_c is the batch prototype (batch feature mean) of class c and N_b is the number of P^b_c in the batch. $\mathcal{Y}_b$ is the label set of samples in the batch. Note that the set of P^b_c and $\mathcal{Y}_b$ are different in different batches. The loss of batch prototype regularization is formulated as follows:

$$\mathcal{L}_{bpr} = -\frac{1}{N_b}\sum_{i=1}^{N_b} log(d^c_i), \tag{7}$$

where d^c_i is the c-th element of d_i and c denotes the ground-truth class of the i-th batch prototype.

3.4 Global Information Distillation

During local updates, we aim to leverage the global model to transfer its knowledge to the local model, ensuring consistency between local and global models. Classifier output contains two components: target class information and non-target class information. Conventional knowledge distillation methods tend to overlook the significance of non-target class information, which is crucial for

effective distillation. Acknowledging the importance of non-target class information is particularly vital in federated scenarios with limited client data access. The global classification information within non-target classes is pivotal in mitigating classifier bias. Inspired by decoupled knowledge distillation (DKD) [47], which separates traditional knowledge distillation into Target Class Knowledge Distillation (TCKD) and Non-Target Class Knowledge Distillation (NCKD), we employ decoupled knowledge distillation to mitigate the occurrence of classifier drift resulting from class imbalance. This is achieved by enabling the local model to acquire a richer understanding of global information through NCKD. The loss of DKD is formulated as:

$$\begin{aligned} DKD(p^{\mathcal{T}}||p^{\mathcal{S}}) &= TCKD + \xi NCKD \\ &= KL(b^{\mathcal{T}}||b^{\mathcal{S}}) + \xi KL(\hat{p}^{\mathcal{T}}||\hat{p}^{\mathcal{S}}), \end{aligned} \tag{8}$$

where $\mathcal{T}$ and $\mathcal{S}$ denote teacher and student respectively. ξ is a parameter that balances the importance of TCKD and NCKD. When ξ is larger, NCKD will have greater impact, and the clients will learn more knowledge that they cannot access. When a sample belonging to class c is input, the probability output by the teacher model can be denoted as $p^{\mathcal{T}} \in \mathbb{R}^{1\times C}$, and the probability output by the student model can be denoted as $p^{\mathcal{S}}$. $b = [p_c, 1-p_c]$ denotes the binary probability of the target class p_c and all other non-target classes $(1-p_c)$; $b^{\mathcal{T}}$ and $b^{\mathcal{S}}$ represent the teacher's and student's binary probability, respectively. We propose $\hat{p} = [\hat{p}_1, \ldots, \hat{p}_{c-1}, \hat{p}_{c+1}, \ldots, \hat{p}_C] \in \mathbb{R}^{1\times(C-1)}$ to construct probabilities of non-target classes, where $\hat{p}^{\mathcal{T}}$ and $\hat{p}^{\mathcal{S}}$ denote teacher's and student's probabilities among non-target classes, respectively. The loss of global information distillation is defined as follows:

$$\mathcal{L}_{gid} = \frac{1}{N} \sum_{z^k_{i,c,t} \in \mathcal{D}^{k+}} \mathrm{I}_{c,t} DKD(p^g_{i,c,t}||p^l_{i,c,t}), \tag{9}$$

where $p^g_{i,c,t} = D(z^k_{i,c,t}|\phi) \cdot W$ and $p^l_{i,c,t} = D(z^k_{i,c,t}|\phi_k) \cdot W$ represent the predicted probabilities output by the global model and the local model, respectively.

3.5 Debiased Classifier Learning

In order to tackle the problem of classifier shift on class-imbalanced clients, we present a debiased classifier learning loss $\mathcal{L}_{dcl}$. Our method involves leveraging fine-grained global prototypes to rebalance the distribution of client samples. By utilizing these prototypes, we can effectively make use of global distribution information to reduce bias in the classifier. The loss is defined as follows:

$$\mathcal{L}_{dcl} = \frac{1}{N} \sum_{z^k_{i,c,t} \in \mathcal{D}^{k+}} \mathrm{I}_{c,t} \mathcal{L}_{CE}(D(z^k_{i,c,t}|\phi_k) \cdot W, c), \tag{10}$$

where $\mathcal{L}_{CE}$ denotes the cross-entropy loss. ϕ_k is the parameters of the k-th local classifier. c is the ground-truth label of $z^k_{i,c,t}$.

Finally, integrating the objectives above, the complete loss function of FedCD for client optimization is formulated as follows:

$$\mathcal{L}_{total} = \mu\mathcal{L}_{dcl} + (1-\mu)\mathcal{L}_{hpc} + \gamma\mathcal{L}_{bpr} + \lambda\mathcal{L}_{gid}, \tag{11}$$

where $\mu = 1 - \frac{t}{T}$. t is the current communication round, and T is the total communication round. During the early training phase, when the model's feature extraction capability is limited, the usage of μ prevents the prototypes from being influenced by inferior features.

Algorithm 1 Local update in FedCD

Input: local epochs E, batch size B,hyperparameter λ, γ, α, M, the datasets of k-th client $\mathcal{D}_k$, the fine-grained global prototypes $\{\mathrm{P}^g\} \rightarrow \{\mathrm{P}^k\}$, the parameter of global model $w^t \rightarrow w^t_k$, current communication round t

```
for epoch e = 1, 2, ..., E do
    for each batch b = {z, y} of D_k^+ do
        Calculate L_hpc in Eq.(5),
        Calculate L_bpr in Eq.(7),
        Calculate L_gid in Eq.(9),
        Calculate L_dcl in Eq.(10),
        Update the local model by optimizing L_total in Eq.(11),
    end for
end for
return w_k^{t+1}, {P^k} to Server
```

Algorithm 2 Server aggregation in FedCD

Input: client number K, fine-grained local prototypes $\{\mathrm{P}^k\}$, parameters of local models $\{w^t_k\}$, total communication round T

```
for each round t = 1, 2, ..., T do
    for each client i = 1, 2, ..., K in parallel do
        w_k^{t+1}, {P^k} ← LocalUpdate(k, w^t, {P^g}),
    end for
    Calculate {P^g} in Eq.(1),
    Aggregate θ^{t+1} as FedAvg,
    Aggregate φ^{t+1} in Eq.(12) - Eq.(15),
    Send w^{t+1} = (θ^{t+1}; φ^{t+1}) and {P^g} to clients,
end for
return w^T
```

In the later stages of training, μ is employed to prevent model overfitting. γ and λ are hyperparameters to adjust the weight of different losses.

3.6 Adaptive Class-Level Classifier Aggregation

In non-IID federated learning, class imbalance on clients leads to varying performance across different classes. Consequently, the importance of different classifiers on a single client differs during the aggregation stage of the global classifier. We propose adaptive class-level classifier aggregation, which mitigates the adverse effects of low-quality classes on the global classifier. Our approach effectively addresses classifier bias by recognizing the varying significance of classes within clients, rather than assigning a uniform weight to each client. Specifically, the parameter of the classifier ϕ can be divided into $C \times M$ class-vectors.

$$\phi = [\varphi^1, \ldots, \varphi^j, \ldots, \varphi^{C\times M}], \tag{12}$$

where the parameters of classifier is $\phi \in \mathbb{R}^{d\times(C\times M)}$ and each class-vector is $\varphi \in \mathbb{R}^{d\times 1}$. d is the dimension of feature representation, and $C \times M$ is the number of fine-grained classes.

We employ the classification outcomes of fine-grained global prototypes, output by the client classifiers, to assess the importance of a particular class among the clients. The expression below illustrates the classification results for fine-grained global prototype of

the j-th class, generated by the classifier D_k from the k-th client:

$$\tilde{\mathrm{p}}_k^j = \sigma(D_k(\mathrm{P}_j^g|\phi_k)), \quad 1 \le k \le K, \tag{13}$$

where $\tilde{\mathrm{p}}_k^j \in \mathbb{R}^{1\times(C\times M)}$ is the classification result, ϕ_k is the parameter of D_k received by the server. Here, $j \in [1, C \times M]$ represents the fine-grained class.

Then, we obtain the weights of the j-th class-vector for all clients as follows:

$$v^j = \sigma([\tilde{\mathrm{p}}_1^j[j], \ldots, \tilde{\mathrm{p}}_k^j[j], \ldots, \tilde{\mathrm{p}}_K^j[j]]), \tag{14}$$

where $\tilde{\mathrm{p}}_k^j[j]$ is the j-th dimension of $\tilde{\mathrm{p}}_k^j$, which indicates the importance of the k-th client on the j-th class. $v^j \in \mathbb{R}^K$ contains the weight of the j-th class vector for each client.

Finally, to obtain an unbiased and reliable global classifier, we perform adaptive class-level global classifier aggregation based on the weight of each client's class-vector as follows:

$$\varphi^j = \sum_{k=1}^{K} v^j[k] \cdot \varphi_k^j. \tag{15}$$

The entire training processes of FedCD on the client and server is presented in Algorithm 1 and Algorithm 2, respectively.

3.7 Discussion on Privacy Protection

A prototype can be considered a low-dimensional representation compared to the original sample. Unlike the original data, the prototype vector contains relatively less information, thus reducing the risk of privacy leakage [35, 39]. The prototype in FedCD is acquired by using gradient descent on the low-dimensional representation of samples belonging to the same class. As a result, our prototype is not merely a linear combination of sample features. This indicates that the process of generating the prototype is irreversible, preventing an attacker from reconstructing the original data based solely on the prototype, unless they have access to the local model. Moreover, FedCD can be effectively combined with other privacy enhancement techniques to further improve the level of privacy protection in the field of federated learning.

4 EXPERIMENTS

4.1 Experimental Settings

Compared methods. We adopt several representative federated learning methods such as FedAvg [33], FedProx [29], Scaffold [23], MOON [26], FedDyn [1], FedDC [15] and FedProc [35] as the comparison methods. To be fair, all these methods adopt the same network architecture and settings.

Datasets. Both CIFAR10 and CIFAR100 datasets [24] have 50,000 training images and 10,000 testing images, the number of classes are 10 and 100 respectively. EMNIST dataset [7] contains 124,800 training images and 20,800 testing images with 26 classes. To obtain non-IID data distribution on clients, we follow the same settings as [26, 46] to partition data by the Dirichlet distribution $\mathbf{Dir}(\beta)$. In the experiments, we set β to 0.5 and 5, where smaller β indicates greater class imbalance.

Network Architecture. For CIFAR10 and EMNIST, we use two convolutional layers followed by maxpooling and two fully connected layers with ReLU activation as the base feature extractor. For CIFAR100, Resnet50 [19] is adopted as the base feature extractor. For all datasets, two fully connected layers are used as the projection, one fully connected layer as the classifier. Slightly different from other methods, our method partitions the classifier into a fine-grained classifier and an aggregation matrix W. This partitioning enables the acquisition and aggregation of fine-grained classification results. The feature extractor consists of the base feature extractor and the projection head. The dimension of the output of the feature extractor z is set to 256.

Hyperparameters. We use the SGD optimizer with a learning rate of 0.01, momentum set to 0.9 and weight decay set to 10^{-5}. The default number of local epochs $E = 10$, the communication round $T = 100$, client number $K = 10$ with the participating rate $\eta = 1$, and the batch size $B = 64$. During local updating, $\{\alpha, \tau, M\}$ is set to $\{0.1, 1, 3\}$ for all datasets, ξ is set to 0.5 for CIFAR10 and 8 for CIFAR100 and EMNIST. $\{\lambda, \gamma\}$ in Eq.11 are set to $\{0.1, 0.1\}$ for all datasets.

4.2 Performance Comparison

Accuracy Comparison. All methods are tested on three benchmark datasets with varying degrees of class unbalance $\beta \in \{0.5, 5\}$ and *iid*. The results in Table 1 demonstrate that FedCD effectively mitigates classifier bias caused by class imbalance and outperforms other methods. Specifically, FedCD achieves at least 3.17%, 4.12%, and 3.96% improvements over all baseline methods for $\beta \in \{0.5, 5\}$ and *iid* on the CIFAR100 dataset. FedProc incorporates prototype learning to enhance client representation learning and demonstrates promising results in various scenarios, affirming the efficacy of prototype learning in federated learning. However, a single prototype cannot precisely capture the distribution information of each class, resulting in inferior performance of FedProc compared to FedCD in all instances, validating the effectiveness of the proposed method FedCD.

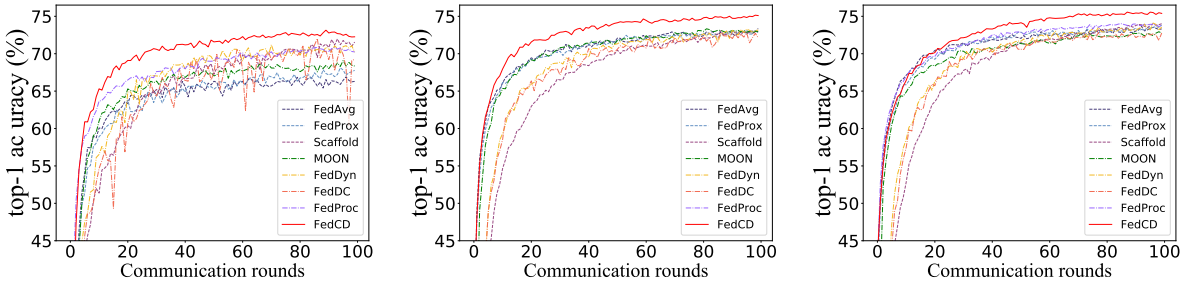
Communication cost Comparison. Table 2 evaluates the communication cost required for all the methods on CIFAR10 and CIFAR100 ($\beta = 0.5$) to achieve the same accuracy as FedAvg with 100 rounds (66.31% for CIFAR10 and 64.50% for CIFAR100). Param represents the parameters transmitted during each round of communication. We calculate multiples of communication rounds and total parameters for all methods relative to FedCD. FedCD achieves the best results on both CIFAR10 and CIFAR100. The total parameters are calculated as the product of the transmitted parameters per round and the number of communication rounds needed to achieve the desired accuracy. We report the number of parameters transmitted by FedCD and FedProc in the form of model parameters plus prototype parameters. The findings indicate that the number of parameters required for transmitting prototypes is significantly smaller compared to model parameters. Additionally, FedCD's total parameters required to achieve the desired accuracy are lower than those of all other methods. Notably, the less optimal approach (FedProc) on CIFAR10 requires 1.5 times more rounds and 1.3 times more total parameters than FedCD, respectively.

4.3 Performance Analysis in Different Settings

To further illustrate the superiority of the proposed FedCD, we conduct experiments from the following aspects to verify the performance of FedCD:

Table 1: Test accuracy (%) of different FL methods on CIFAR10, CIFAR100 and EMNIST.

Method		FedAvg	FedProx	Scaffold	MOON	FedDyn	FedDC	FedProc	**FedCD**
CIFAR10	*iid*	73.39 ± 0.12	73.56 ± 0.05	73.89 ± 0.08	72.70 ± 0.22	73.56 ± 0.29	72.64 ± 0.10	73.77 ± 0.14	**75.56 ± 0.12**
	$\beta = 5$	72.99 ± 0.03	72.81 ± 0.26	72.87 ± 0.12	72.83 ± 0.13	73.04 ± 0.10	72.33 ± 0.28	72.87 ± 0.33	**75.10 ± 0.06**
	$\beta = 0.5$	66.31 ± 0.59	67.53 ± 0.60	71.37 ± 0.25	68.55 ± 0.54	70.68 ± 0.36	69.17 ± 0.08	70.07 ± 0.36	**73.10 ± 0.22**
CIFAR100	*iid*	65.51 ± 0.05	66.03 ± 0.10	56.53 ± 0.12	68.59 ± 0.05	59.94 ± 0.23	65.78 ± 0.29	65.36 ± 0.11	**72.55 ± 0.05**
	$\beta = 5$	65.30 ± 0.16	65.35 ± 0.05	56.17 ± 0.45	68.29 ± 0.20	58.40 ± 0.04	65.19 ± 0.10	64.48 ± 0.09	**72.41 ± 0.08**
	$\beta = 0.5$	64.50 ± 0.24	64.56 ± 0.31	52.50 ± 0.30	67.95 ± 0.15	57.24 ± 0.23	63.08 ± 0.07	63.53 ± 0.13	**71.12 ± 0.03**
EMINST	*iid*	90.86 ± 0.06	91.10 ± 0.19	91.50 ± 0.30	90.65 ± 0.02	89.76 ± 0.17	91.79 ± 0.28	93.11 ± 0.55	**93.70 ± 0.02**
	$\beta = 5$	91.32 ± 0.18	91.04 ± 0.11	91.79 ± 0.23	90.70 ± 0.06	89.44 ± 0.38	91.64 ± 0.19	92.84 ± 0.16	**93.58 ± 0.02**
	$\beta = 0.5$	90.65 ± 0.09	91.09 ± 0.08	90.44 ± 1.20	90.83 ± 0.02	90.06 ± 0.18	91.57 ± 0.31	92.29 ± 0.24	**92.49 ± 0.05**

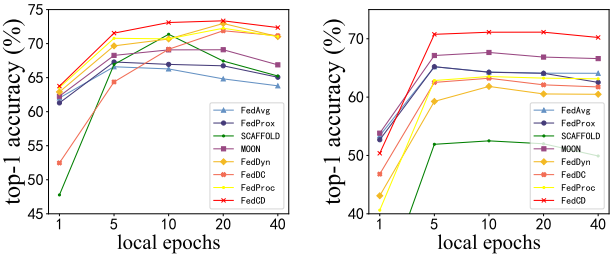


(a) $\beta = 0.5$ (b) $\beta = 5$ (c) **iid**

Figure 3: Learning curve under different degree of class imbalance. (a) $\beta = 0.5$, (b) $\beta = 5$, (c) *iid* on CIFAR10 dataset

Table 2: The communication cost needed for various FL methods to achieve the same accuracy as 100 rounds of FedAvg on CIFAR10 and CIFAR100 ($\beta = 0.5$).

	CIFAR10			CIFAR100		
	rounds	param(MB)	total(MB)	rounds	param(MB)	total(MB)
FedAvg	100 (7.6×)	3.53	353 (6.7×)	100 (3.3×)	1077.5	107750 (3.3×)
FedProx	52 (4×)	3.53	183.56 (3.5×)	75 (2.5×)	1077.5	80812.5 (2.4×)
Scaffold	36 (2.7×)	7.06	254.16 (4.8×)	\	2155	> 215500
MOON	29 (2.2×)	3.53	102.37 (1.9×)	43 (1.4×)	1077.5	46332.5 (1.4×)
FedDyn	25 (1.9×)	3.53	88.25 (1.6×)	\	1077.5	> 107750
FedDC	28 (2.1×)	7.06	197.68 (3.7×)	\	2155	> 215500
FedProc	20 (1.5×)	3.53+0.09	72.4 (1.3×)	\	1077.5+0.97	> 107847
FedCD	**13** (1×)	3.72+0.29	**52.13** (1×)	**30** (1×)	1079.4+2.9	**32469** (1×)



(a) **CIFAR10** (b) **CIFAR100**

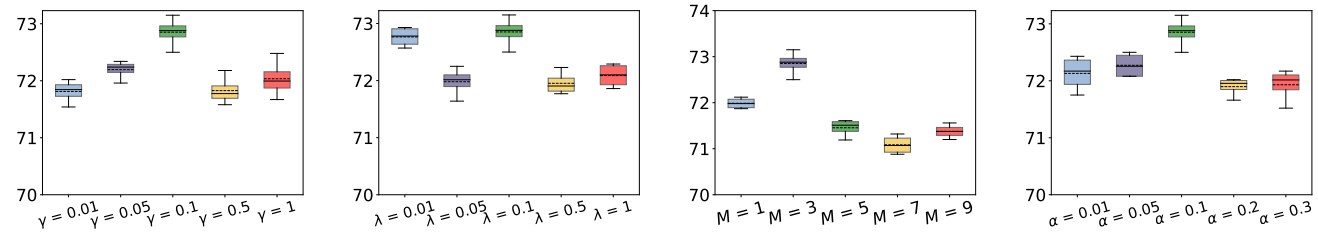
Figure 4: The accuracy of all methods on local epochs.

Degree of Class Imbalance. Figure 3 displays the learning curves of all methods on CIFAR10 under different degrees of class imbalance, where FedCD achieves the best results in $\beta = 0.5$, $\beta = 5$, and *iid* scenarios respectively. As shown in Figure 3(a) and 3(b), FedCD exhibits rapid and steady growth from the beginning, and its final accuracy is also higher than that of other methods. Figure 3(c) reveals that although the initial growth rates of MOON, FedProc and FedAvg are large, FedCD's accuracy surpasses them after 20 rounds, demonstrating the effectiveness of our method.

Table 3: The accuracy (%) of all methods with different numbers of clients and varying communication rounds on CIFAR100.

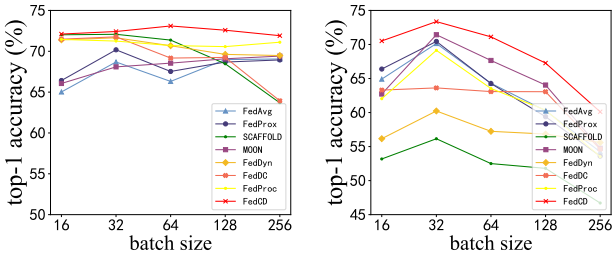
Method	Client Number=50		Client Number=100	
	100 rounds	200 rounds	250 rounds	500 rounds
FedAvg	51.4	55.8	51.0	55.0
FedProx	51.3	56.2	51.3	54.6
Scaffold	35.3	43.6	37.4	44.5
Moon	57.9	63.0	56.9	61.8
FedDyn	52.0	56.8	53.5	55.3
FedDC	53.2	58.4	54.2	57.3
FedProc	54.6	60.8	55.7	59.6
FedCD	**59.7**	**66.3**	**59.43**	**66.31**

Various Local Epochs and Batch Sizes. We investigate the influence of local epoch and batch size on the overall performance of the global model. The findings are depicted in Figure 4 and Figure 6. The accuracy of SCAFFOLD is too low to display when the number of local epochs is set to 1 on CIFAR100 (20.4%). When the local epoch is set to 1 on the CIFAR100 dataset, both FedCD and FedProc struggle to learn accurate prototypes due to the dataset's numerous classes and insufficient local update epochs. Consequently, FedCD does not yield optimal results. Moreover, FedCD performs well when the local epoch is large, indicating its effectiveness in mitigating classifier bias. On CIFAR100, increasing the batch size leads to a decrease in the number of local update rounds, resulting in a somewhat diminished model performance. This phenomenon is observed across all methods. Nevertheless, FedCD consistently achieves superior outcomes for all batch sizes.



(a) **Box plot w.r.t.** γ (b) **Box plot w.r.t.** λ (c) **Box plot w.r.t.** M (d) **Box plot w.r.t.** α

Figure 5: Parameter sensitivity analysis for λ, γ, M and α on CIFAR10 dataset with $\beta = 0.5$.



(a) **CIFAR10** (b) **CIFAR100**

Figure 6: The accuracy of all methods under different batch size.

Different Number of Clients. To assess the scalability of FedCD, we test it with a larger number of clients on the CIFAR100 dataset. We set the number of clients to 50 and 100, respectively, with only 20% of the clients participating in federated learning in each round ($\eta = 0.2$). For example, when the number of clients is set to 100, we randomly select 20 clients to participate in the local update before each communication and then aggregate these 20 models as the global model. The results are shown in Table 3. For 100 clients, compared with Moon, FedCD achieves 2.53% higher accuracy at 250 rounds, and achieves 4.51% higher accuracy at 500 rounds. The results demonstrate the scalability of FedCD when a large number of clients participate in federated learning.

4.4 Parameter Sensitivity and Ablation Study

The sensitivity of four important parameters γ, λ, M and α in FedCD are studied. We use the CIFAR10 dataset for parameter sensitivity analysis, selecting λ and γ from $[0.01, 0.05, 0.1, 0.5, 1]$, M from $[1, 3, 5, 7, 9]$ and α from $[0.01, 0.05, 0.1, 0.2, 0.3]$. Figure 5 illustrates the accuracy of FedCD under different parameters with box plots. We can observe that the performance of FedCD remains relatively stable across a wide range of parameter variations, indicating that our method is not sensitive to parameters.

Ablation experiments are performed on CIFAR10 and CIFAR100, respectively. To assess the efficacy of each component in FedCD, we introduce six degraded methods. (1) **FedCD-hpc**: Remove hierarchical prototype contrastive learning from FedCD and update fine-grained local prototypes by computing feature means. (2) **FedCD-dkd**: Remove global information distillation loss from FedCD. (3) **FedCD-bpr**: Remove batch prototype regularization loss from FedCD. (4) **FedCD-acla**: Remove the adaptive class-level aggregation and aggregate global model like FedAvg. (5) **FedCD/kd**: Replace global information distillation loss with KL divergence

Table 4: Ablation study of FedCD in terms of accuracy (%).

Method	CIFAR10	CIFAR100
FedCD	**73.1**	**71.12**
FedCD-hpc	71.74	68.01
FedCD-dkd	72.25	70.24
FedCD-bpr	72.08	69.99
FedCD-acla	72.05	69.61
FedCD/kd	72.25	70.3
FedCD/cla	72.34	70.14

substitution loss. (6) **FedCD/cla**: Employ the ratio of fine-grained samples number on the client to the total number of fine-grained samples as a weight to replace the adaptive weight for aggregating fine-grained global classifiers. The experimental results of ablation study are shown in Table 4. FedCD achieves superior performance compared to each degraded method. The ablation experiment results reveal that by integrating all components of FedCD, we establish a federated learning framework that effectively tackles the classifier bias issue in class-imbalanced federated learning.

5 CONCLUSION

In this paper, we introduce a framework to address classifier bias in non-IID federated learning. Our proposed method integrates fine-grained prototype learning, global information knowledge distillation, and adaptive class-level classifier aggregation within a unified framework. By correcting classifier bias at different stages of federated learning including representation and classifier learning on clients, as well as global aggregation on server, our method outperforms existing non-IID federated learning methods. Notably, we introduce hierarchical prototype contrastive learning, enabling the acquisition of fine-grained prototypes for each class, which better captures the global distribution compared to using a single prototype. We leverage global information distillation to decouple global prediction information, effectively correcting local classifier bias. Furthermore, our framework discriminates the quality of class vectors in classifiers, mitigating the impact of classifier bias during the aggregation stage. Extensive experiments demonstrate the superior performance of FedCD compared to state-of-the-art non-IID federated learning methods.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (62272058, 62192784, U22B2038, 62172056) and CCF-Tencent Open Research Fund.

REFERENCES

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. 2021. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263* (2021).

[2] Durmus Alp Emre Acar, Yue Zhao, Ruizhao Zhu, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. 2021. Debiasing model updates for improving personalized federated training. In *International Conference on Machine Learning*. PMLR, 21–31.

[3] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2021. Provably secure federated learning against malicious clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6885–6893.

[4] Hong-You Chen and Wei-Lun Chao. 2020. Fedbe: Making bayesian model ensemble applicable to federated learning. *arXiv preprint arXiv:2009.01974* (2020).

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[6] Xiaomeng Chen, Yingxia Shao, Zhe Xue, and Ziqiang Yu. 2021. Multi-modal COVID-19 discovery with collaborative federated learning. In *2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS)*. IEEE, 52–56.

[7] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. 2017. EMNIST: Extending MNIST to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2921–2926.

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

[9] Don Kurian Dennis, Tian Li, and Virginia Smith. 2021. Heterogeneity for the win: One-shot federated clustering. In *ICML*. PMLR, 2611–2620.

[10] Kate Donahue and Jon Kleinberg. 2021. Model-sharing games: Analyzing federated learning under voluntary participation. In *AAAI*, Vol. 35. 5303–5311.

[11] Haya Elayan, Moayad Aloqaily, and Mohsen Guizani. 2021. Deep federated learning for IoT-based decentralized healthcare systems. In *2021 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 105–109.

[12] Haya Elayan, Moayad Aloqaily, and Mohsen Guizani. 2021. Sustainability of healthcare data analysis IoT-based systems using deep federated learning. *IEEE Internet of Things Journal* 9, 10 (2021), 7338–7346.

[13] Angelo Feraudo, Poonam Yadav, Vadim Safronov, Diana Andreea Popescu, Richard Mortier, Shiqiang Wang, Paolo Bellavista, and Jon Crowcroft. 2020. CoLearn: Enabling federated learning in MUD-compliant IoT edge networks. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*. 25–30.

[14] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. 2021. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In *ICML*. PMLR, 3407–3416.

[15] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. 2022. FedDC: Federated Learning with Non-IID Data via Local Drift Decoupling and Correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10112–10121.

[16] Zeli Guan, Yawen Li, Zhe Xue, Yuxin Liu, Hongrui Gao, and Yingxia Shao. 2021. Federated graph neural network for cross-graph node classification. In *2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS)*. IEEE, 418–422.

[17] Sungwon Han, Sungwon Park, Fangzhao Wu, Sundong Kim, Chuhan Wu, Xing Xie, and Meeyoung Cha. 2022. FedX: Unsupervised Federated Learning with Cross Knowledge Distillation. *arXiv preprint arXiv:2207.09158* (2022).

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[20] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335* (2019).

[21] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2020. Federated visual classification with real-world data distribution. In *ECCV*. Springer, 76–92.

[22] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. 2021. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7865–7873.

[23] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*. PMLR, 5132–5143.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[25] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. 2021. Adaptive Prototype Learning and Allocation for Few-Shot Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8334–8343.

[26] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10713–10722.

[27] Qiushi Li, Wenwu Zhu, Chao Wu, Xinglin Pan, Fan Yang, Yuezhi Zhou, and Yaoxue Zhang. 2020. InvisibleFL: federated learning over non-informative intermediate updates against multimedia privacy leakages. In *Proceedings of the 28th ACM International Conference on Multimedia*. 753–762.

[28] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*. PMLR, 6357–6368.

[29] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* 2 (2020), 429–450.

[30] Yawen Li, Wenling Li, and Zhe Xue. 2022. Federated learning with stochastic quantization. *International Journal of Intelligent Systems* 37, 12 (2022), 11600–11621.

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

[32] Yongqiang Ma, Yingxia Shao, Zhe Xue, and Ziqiang Yu. 2021. Urban Fatigue Driving Prediction With Federated Learning. In *2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS)*. IEEE, 47–51.

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

[34] Umberto Michieli and Pietro Zanuttigh. 2021. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *CVPR*. 1114–1124.

[35] Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. 2023. Fedproc: Prototypical contrastive federated learning on non-iid data. *Future Generation Computer Systems* 143 (2023), 93–104.

[36] Qiying Pan, Yifei Zhu, and Lingyang Chu. 2023. Lumos: Heterogeneity-aware Federated Graph Learning over Decentralized Devices. *IEEE International Conference on Data Engineering (ICDE)* (2023).

[37] Yu Qiao, Md Munir, Apurba Adhikary, Huy Q Le, Avi Deb Raha, Chaoning Zhang, Choong Seon Hong, et al. 2023. MP-FedCL: Multi-Prototype Federated Contrastive Learning for Edge Intelligence. *arXiv preprint arXiv:2304.01950* (2023).

[38] Hao Sun, Yuan Jia, Hui Dong, Dibo Dong, and Jianping Zheng. 2020. Combining additive manufacturing with microfluidics: an emerging method for developing novel organs-on-chips. *Current Opinion in Chemical Engineering* 28 (2020), 1–9.

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

[40] Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen. 2022. FedCor: Correlation-Based Active Client Selection Strategy for Heterogeneous Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10102–10111.

[41] Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10, 3152676 (2017), 10–5555.

[42] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440* (2020).

[43] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems* 33 (2020), 7611–7623.

[44] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[45] Yu Zang, Zhe Xue, Shilong Ou, Yunfei Long, Hai Zhou, and Junping Du. 2023. FedPcf: An Integrated Federated Learning Framework with Multi-Level Prospective Correction Factor. In *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*. 490–498.

[46] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. 2022. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10174–10183.

[47] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. 2022. Decoupled Knowledge Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11953–11962.

[48] Jian Zhao, Jianshu Li, Xiaoguang Tu, Fang Zhao, Yuan Xin, Junliang Xing, Hengzhu Liu, Shuicheng Yan, and Jiashi Feng. 2019. Multi-prototype networks for unconstrained set-based face recognition. *arXiv preprint arXiv:1902.04755* (2019).