Wuhan University
Journal of Natural Sciences

2022, Vol.27 No.6, 508-520

Article ID 1007-1202(2022)06-0508-13
DOI https://doi.org/10.1051/wujns/2022276508

MpFedcon : Model-Contrastive Personalized Federated Learning with the Class Center

□ LI Xingchen, FANG Zhijun, SHI Zhicai
School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Abstract: Federated learning is an emerging distributed privacy-preserving framework in which parties are trained collaboratively by sharing model or gradient updates instead of sharing private data. However, the heterogeneity of local data distribution poses a significant challenge. This paper focuses on the label distribution skew, where each party can only access a partial set of the whole class set. It makes global updates drift while aggregating these biased local models. In addition, many studies have shown that deep leakage from gradients endangers the reliability of federated learning. To address these challenges, this paper propose a new personalized federated learning method named MpFedcon. It addresses the data heterogeneity problem and privacy leakage problem from global and local perspectives. Our extensive experimental results demonstrate that MpFedcon yields effective resists on the label leakage problem and better performance on various image classification tasks, robust in partial participation settings, non-iid data, and heterogeneous parties.

Key words: personalized federated learning; layered network; model contrastive learning; gradient leakage
CLC number: TP 301

Received date: 2022-08-24
Foundation item: Supported by the Scientific and Technological Innovation 2030—Major Project of "New Generation Artificial Intelligence" (2020AAA 0109300)
Biography: LI Xingchen, male, Master candidate, research direction: federated learning. E-mail: 351977119@qq.com

0 Introduction

Data resources have become the lifeline of modern enterprise value creation and the new engine of digital technology power. In the process of industrial digital transformation, a large amount of valuable data scattered among all parties is generated. Due to increasing privacy concerns and data protection regulations[1], all parties cannot send their private data to a central server to train models. Federated learning (FL) is an emerging distributed machine learning paradigm that uses decentralized data from multiple parties to jointly train a shared global model without sharing the individuals' raw data[2-6]. FL has achieved remarkable success in various industrial applications such as autonomous driving[7], wearable devices[8], medical diagnostics[9,10], and cell phones[11, 12]. However, the non-independent identically distributed (non-iid) data poses a significant challenge. The data distribution of parties in FL might be highly variable since parties separately collect local data based on their preferences and sampling space. Label distribution skew is a common and serious category of non-iid[3]. Some studies have proved that the non-iid data causes drift in the local updates of parties[13,14]. In addition, the global model is further scattered by a collection of mismatched local optimal solutions, which eventually leads to a slow and unstable convergence of the overall training process[15-17].

A variety of efforts attempt to address non-iid data challenges. Some studies have shown that reducing data variability can improve the convergence of the global FL model[18,19]. However, they usually need to modify the lo-

cal distribution, which might result in the loss of important data about the inherent diversity of consumer behavior. Some methods stabilize the local training phase by adjusting the local and global model deviation across the parameter space, such as FedProx[20], SCAFFOLD[13]. Other studies such as Ditto[19], APFL[21] improve the generalization ability of the model by mixing global and local model strategies. We admit the fact that the local optimal points of parties are fundamentally inconsistent with the global optimal point in the heterogeneous FL setup. The majority of prior FL methods, however, compel local models to be consistent with the global model and ignore the problem of privacy leakage. For instance, DLG[22] and iDLG[23] have revealed that existing gradient-based privacy breaches are mainly attacked by inference through the properties of the last layer of the neural network.

Based on the above inference, we propose a model-contrastive personalized learning with the class center, dubbed as MpFedcon, which is a typical personalized federated learning framework based on FedAvg (FederatedAveraging). Specifically, we apply a layered network that decouples the target neural network into a base encoder that participates in collaborative training and a locally preserved personalization layer. The base encoder layer learns global knowledge, while the personalization layer retains sensitive information to resist the deep leakage of gradients. Each party's local training is corrected from a global perspective by using the global class center contrastive learning. A global class center is defined as each class's average vector of representations[24]. Further, inspired by Simsiam[25], MpFedcon greatly reduces the computational complexity by using only positive samples for training rather than negative sample pairs and large batches through model-contrastive learning[26]. MpFedcon significantly outperforms the other state-of-the-art federated learning algorithms on various image classification datasets, including CIFAR-10, CIFAR-100, and FEMNIST[25,27]. For instance, MpFedcon achieves 83.3% top-1 accuracy on FEMNIST with 100 parties, while the best top-1 accuracy of existing studies is 78.56%. Compared with the most classic FedAvg in the non-iid setting, MpFedcon improves the convergence speed by 3.7 and 28.5 times and reduces the communication cost by 73.2% and 96.5% on the CIFAR-10 and CIFAR-100, respectively. The rest of this paper is arranged as follows: Section 1 reviews the related work of the FL, contrastive learning, and leakage from gradients. Section 2 explores the influence of local drift in FL. Section 3 gives problem statement and motivation. Section 4 describes the proposed method. The experimental results are presented in Section 5 to demonstrate the efficiency of our method. Finally, Section 6 concludes our work. Overall, the main contributions of this paper are as follows:

1) We propose a new personalized federated learning framework to solve the label distribution skew in FL, which mitigates the local and global drift problem by introducing the global class center model-contrastive learning to correct local training.

2) We explore the causes of gradient-based privacy leakage, then design and verify the effectiveness of layered networks for defending against gradient leakage attacks.

3) We design the local layered network architecture to effectively learn the global underlying knowledge through supervised loss and contrastive loss functions, which promotes tight intra-class and separable inter-class sample sets in the classification space.

4) We implement MpFedcon and conduct extensive experiments on different datasets. The results show that MpFedcon outperforms state-of-the-art methods regarding inference accuracy and computational efficiency.

1 Related Work

1.1 Federated Learning

The standard federated learning approach aims at learning a single shared model that performs well on average across all parties. The classical federated learning method FedAvg[4] follows the typical four-step protocol shown in Fig. 1. ① The server randomly initializes the parameters of the global model and sends them to each party. ② Upon receiving the global model, each party updates the model based on its local training data using stochastic gradient descent (SGD). ③ The selected party uploads its local model parameters back to the server. ④ The server averages the model parameters to generate the global model for the next training round. Repeating these steps until convergence.

The non-iid problem has been addressed in a wealth of studies with three main aspects: local training improvements, aggregation, and personalized models. Improvements in local training such as FedProx[20] proposed to add a proximal term to normalize the Euclidean distance between the local and global models.

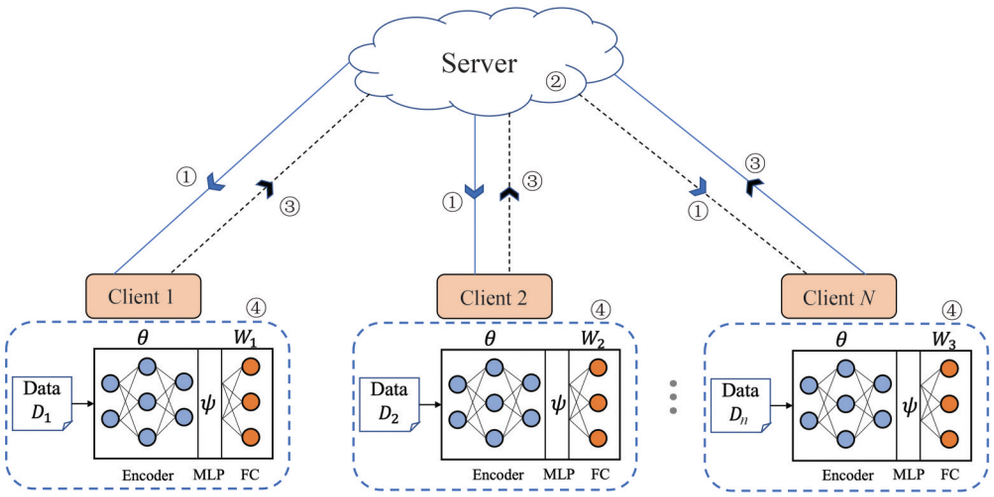

Fig. 1 The process of federated learning process

① Transfer model parameters; ② Security aggregation; ③ Uploading local parameters; ④ Local training; MLP: Multilayer perceptron; FC: Fully connected

SCAFFOLD[13] corrected the drift in local updates by introducing control variables. Other works were to improve aggregation efficiency, such as FedNova[18]. APFL[21] explored adaptive adjustment of global and local models to achieve personalized models. Fedper[28], Fedrep[29], and others explored layered network architecture, which aims to train personalized models for individual parties rather than a shared global model.

1.2 Contrastive Learning

The core idea of contrastive learning is to attract positive and reject negative sample pairs. Contrastive learning is widely used in self-supervised representation learning. Supervised contrast learning is an extension of contrastive learning by combining label information to compose positive and negative samples. In fact, contrastive learning methods benefit from generous negative samples. InfoDist[30] uses a memory bank to store negative sample pairs. SimCLR[31] directly uses the negative samples coexisting in the current batch, so it requires a large batch size. However, selecting representative and informative negative samples is a critical and challenging task. SimSiam[25] proposes a simple twin network to learn representations without negative sample pairs, large batch, and momentum encoding.

Contrastive learning in federated learning has recently emerged as an effective approach to solving non-iid problems. Some existing approaches use a contrastive loss to compare different image representations, and they can utilize the huge unlabeled data on distributed edge devices[32, 33]. Wang *et al*[34] used a supervised contrastive learning to improve the quality of learned features to solve the long-tail distribution problem in classification tasks. Wang *et al*[35] explored the application of contrastive federated learning in medical image segmentation. However, they ignored the need for personalized models and did not explore the issue of gradient-based privacy leakage. In contrast to previous work, we introduce model-contrastive learning with the global class center into supervised learning to address the issues of inconsistency in the embedding space for each party.

1.3 Leakage from Gradients

It is generally accepted that exchanging gradients across parties will not leak private training data in distributed learning systems, such as collaborative learning[36] and federated learning[2, 3]. Recently, Zhu *et al*[22] proposed a method called DLG, which shows the possibility of obtaining private training data from publicly shared gradients. DLG[22] synthesizes virtual data and corresponding labels under the supervision of shared gradients. The iDLG[23] further demonstrates that the last layer of shared gradients must leak ground truth labels when the activation function is non-negative. Wainakh *et al*[37] further explored the properties of gradient-based leakage of true labels under large batch. Common techniques for protecting privacy include adding noise, gradient compression, discretization, and differential privacy-preserving. But all these methods reduce the model accuracy to different degrees.

2 Local Drift in Federated Learning

In FedAvg, all parties optimize their models on the local dataset for each training round. Then the server up-

dates the global model based on the expectations of the local model parameters. The objective is to solve:

$$w^* = \arg\min_w L(w) = \frac{1}{n}\sum_{i=1}^{n}\frac{|D_i|}{|D|} f_i(w) \tag{1}$$

where n is the number of parties, D_i is the private local dataset of party i, and $f_i(w)$ is the expected loss of party i. The overall goal is to obtain a globally optimal model w^* on the global dataset $D=\bigcup_{i\in[n]} D_i$.

There is a drift between the local and global models due to the label distribution skew, a special kind of non-iid scene, where each party can only access a partial set of the whole class set[38]. The performance of FedAvg is significantly reduced with the highly skewed non-iid data in FL[13,20,39], indicating that ignoring local drift results in the deviation of global model. For this purpose, we give a baseline approach called SOLO, in which each party trains the model only by its local data without federated learning. In Fig. 2, we use a simple example to illustrate that a local drift in the party will lead to a biased global model in FedAvg. It assumes that the model has a non-linear transformation function f(e. g., leaky-relu). Suppose w_1 and w_2 are local parameters for party 1 and party 2, x is a data point, and the corresponding outputs for party 1 and party 2 are $y_1=f(w_1,x)$ and $y_2=f(w_2,x)$. The parameters of the model generated by FedAvg can then be expressed as $w_f=w_1+w_2$. w_c is a parameter of the centralized model that can get the ideal output. As shown in Fig. 2, we have $w_f\neq w_c$ and $f(w_f,x)\neq\frac{y_1+y_2}{2}$, indicating that the global model in FedAvg is skewed, which may lead to slow convergence and poor accuracy.

Figure 3 shows the precision results of training using only local data sets and MSE (mean square error) distance between the models in SOLO and FedAvg under the same conditions. It indicates that the accuracy cannot be improved obviously, and the inter-party drift becomes more severe as the number of local iterations increases.

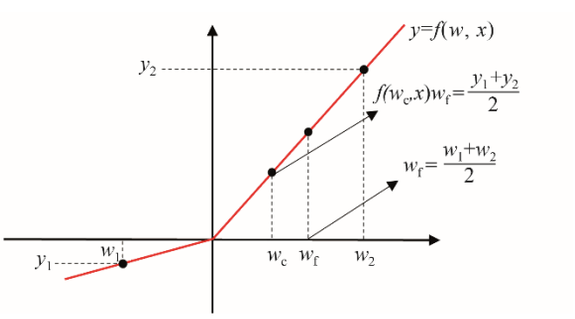

Fig. 2 Illustration of the local drift in FedAvg with a leaky-relu activation

In this case, each party should have a personalized model to suit its unique data distribution. It is necessary to correct the local optimization direction from a global perspective to align the local optimization direction with the global optimization direction to improve the effect of FL.

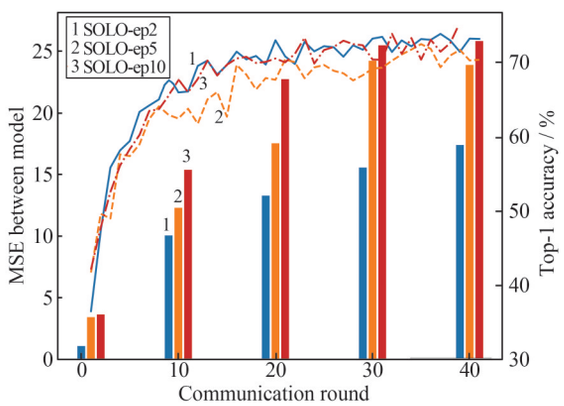

Fig. 3 Impact of different epochs when the party uses only local data

"ep" represents the number of local epochs; The bar chart shows the MSE distance of the SOLO and FedAvg models; The curves indicate the accuracy of the different local epochs for each round

3 Problem Statement and Motivation

Suppose there are N parties $(P_1,\cdots,P_n)$, where party P_i has a local dataset $D_i=\left(x_j,y_j\right)_{j=1}^{N^{(i)}}$. The server and parties attempt to jointly learn the parameters of the global representation, while the party tries to learn its unique model locally. The personalized federated learning can solve:

$$\min_{w\in\mathbb{R}^d} F(w) = \frac{1}{N}\sum_{i=1}^{N}\frac{|D_i|}{|D|} f_i(w_i) \tag{2}$$

where $F(w)=E_{(x,y)\sim D_i}\left[f_i(w,(x,y))\right]$ is the empirical loss of P_i, f_i and w_i are the error function and learning model of the P_i. Most participants do not have sufficient local data and can only observe a subset of the total categories in practical federated learning scenarios. Parties may be unable to obtain solutions with the expected low risk through local training. Therefore, parties need to learn the model through federated learning to use the cumulative data from all parties. MpFedcon is based on an intuitive idea: It can improve the accuracy of classification tasks through correcting local and global distribution consistency in label-absent scenarios in FL; a layered network facilitates the construction of a personalized

model, then personalized layers further fit its data distributions and prevent sensitive information leakage. The effectiveness of layered networks against gradient leakage is analyzed in Section 4.4.

To further verify this intuition, we now discuss the observations that motivate the correction of local training. We explore a more skewed data imbalance issue: label distribution skew, which means each party could only access a subset of the entire class collection[40]. Specifically, we first train a CNN (Convolutional Neural Network) model on CIFAR-10 as a center model. Then, we partition the dataset into 10 subsets in an unbalanced manner and train a CNN model on each subset as SOLO model, where a subset contains 5 classes of data. We use the t-SNE[41] to visualize the hidden vectors of images from a randomly selected SOLO model and center model as shown in Fig. 4(a) and Fig. 4(b). The SOLO method learns better features, but its clustering degree and clustering centers differ significantly from the global distribution in the ideal condition. This may hinder the accuracy of downstream classification tasks. Figure 4(c) shows the representation learned by the FedAvg algorithm. We can observe that the points with the same class are more confused in Fig. 4(c) compared with Fig. 4(a). The FedAvg even leads the model to learn a worse representation due to the skewed local data distribution. This further verifies that the inconsistency of local and global data distribution will significantly affect the performance of federated learning. MpFedcon corrects the local update direction by introducing a global class center from the perspective of global clustering. As shown in Fig. 4(d), the local party data are restricted to the same region as the global distribution after the MpFedcon method, so there is space further to improve the aggregation effect of the central model and enhance the classification effect of downstream tasks.

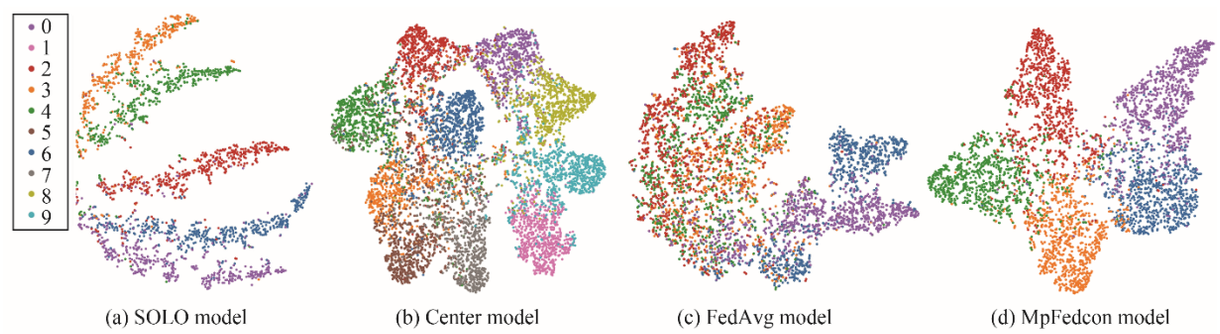

Fig. 4 T-SNE visualizations of hidden vectors on CIFAR-10

4 Method

Based on the above ideas, we propose MpFedcon, a simple and effective FL framework based on FedAvg. Since there is a fundamental contradiction between local and global optimum, MpFedcon aims to constrain the local update direction to be consistent with the global optima, and further fit its unique data distribution by personalized layers while sensitive information is retained locally. In the following, we present the local network architecture, the global class center, the local objective, and privacy protection based on gradient leakage.

4.1 Local Network Architecture

As shown in Fig. 5, the local network consists of three components: a base encoder, a projection head, and an output layer. Specifically, since the heterogeneous data distributed across tasks may share a common representation, we use the base encoder to extract common representation vectors from inputs to improve the quality of each party model. Then the representation is mapped to a space with a fixed dimension using an additional projection head. We use a multilayer perceptron (MLP) with hidden layers to implement the projection head, which helps to improve the representation of the layers that precede it[31]. At last, the output layer predicts values for each class. Locally retained personalized layers include a projection head and an output layer that protect privacy and adapt to local data distribution. It further mitigates the impact of non-iid on model training.

For ease of representation, with model weight w, we use $F_w(\cdot)$, $D_w(\cdot)$, $P_w(\cdot)$ and $O_w(\cdot)$ to denote the entire network, base encoding, projection head, and output layer, respectively. When studying the supervised setup, the base encoder extracts the feature representation from the input x. The feature representation is mapped to the

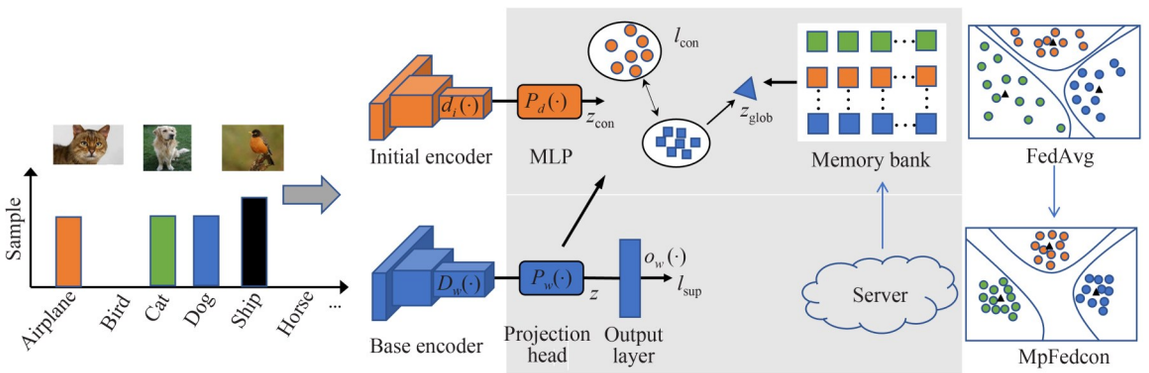

Fig. 5 Overview of *i*-th local network architecture in MpFedcon

The feature extraction network (including the initial encoder, base encoder and MLP) extracts the representation z, z_{con} and then the local network is combined with global center features z_{glob} to calculate the contrast loss l_{con}. The output layer FC predicts the class-wise logits to compute the cross-entropy l_{sup}

low-dimensional space through the projection head for computing the contrast loss l_{con}. The output layer predicts class-wise logits s, which are used to calculate typical loss terms in supervised learning. The model for P_i is composition of its local parameters and the representation: $w_i(x)=\left(h_i \circ d_w\right)(x)$, where h_i is the locally retained personalized layers, including a projection head and output layer, and d_w denotes a common representation of the base encoder extraction.

4.2 The Global Class Center

As shown in Fig. 5, we introduce the global class center as the optimization target for each class from a global perspective. The global server stores and maintains the class centers through a Memory Bank[42]. In the supervised scenario, samples of the same class are restricted to the class center region, thus effectively solving the problem of skewed optimization direction due to the label distribution skew. The classes centers are updated as follows:

$$c_i^t=\frac{1}{m}\sum_{x_i \in \pi} P_w\left(x_i\right) \tag{3}$$

$$c^{t+1} \leftarrow \frac{1}{n}\sum_{i=1}^{n} c_i^t \tag{4}$$

where x_i denotes the samples of class i, m is the number of class samples, $P_w(\cdot)$ denotes the feature output of the projection head, c_i^t is the local class center obtained after training local data in round t. The class center of each party is aggregated and averaged on the server to obtain the global class center c^{t+1}, then the server distributes it to participants next round. Aggregated data is more conducive to training federated learning than skewed data. We aim to find more desirable class center locations from a global perspective and thus improve the classification performance of downstream tasks.

4.3 Local Objective

The local loss consists of two parts. The first part is a typical loss term in supervised learning (e.g., cross-entropy loss), denoted as l_{sup}. The second part is our proposed global class center model contrastive loss term, denoted as l_{con}. In the t-th training round, party i receives a common base encoder model θ^t and the global class center set $M\left(\sum_{i=1}^{m} c_i \in M\right)$, θ^t combined with the locally retained personalized layers h_i as the initialized w_i^t for this round. Let $d_i^t \leftarrow \theta_i^t$, where d_i^t denotes the initial model parameters in this round and does not participate in the gradient update. Let $z_{\text{glob}}=c_i^t$ represent the class center feature vector of the i-th class. $z=P_w(x)$ represents the feature representation from the local model w_i^t being updated, $z_{\text{con}}=P_d(x)$ is the mapped representation of input x by the initial model d_i^t. Since the global model has a more robust representation, we correct the local update direction by reducing z and z_{glob} and increasing the distance between z and z_{con} through the global class center c_i. The model contrastive loss is defined as:

$$l_{\text{con}}=-\log \frac{\exp\left(\frac{\text{sim}\left(z, z_{\text{glob}}\right)}{\tau}\right)}{\exp\left(\frac{\text{sim}\left(z, c_i\right)}{\tau}\right)+\exp\left(\frac{\text{sim}\left(z, z_{\text{con}}\right)}{\tau}\right)} \tag{5}$$

where τ denotes the temperature parameter, $\text{sim}(\cdot,\cdot)$ is the cosine similarity. The local objective is to minimize

$$\min_{w_i^t} {}_{(x,y)\sim D^i} E\left[l_{\text{sup}}\left(w_i^t;(x,y)\right)+\mu l_{\text{con}}\left(w_i^t; w_i^{t-1}; c_i; x\right)\right] \tag{6}$$

where μ is the hyperparameter that regulates the weights of the two terms. The overall algorithm is described in algorithm 1.

Algorithm 1: The MpFedcon framework

Input: number of communication rounds T, number of parties n, number of local epochs E, participation rate r, step size α, temperature τ, learning rate η, hyper-parameter μ, number of local updates for the common representation $\tau_{\varnothing}$, number of local updates for the head τ_{h}

Output: The final model $w_1^t, w_2^t, \cdots, w_n^t$

Server executes:

1.Initialize w^0, c^0

2.Receives a batch of parties of size r, n

3.For $t=1,2,\cdots,T-1$ do

4. send the global base encoder model θ^t to C_i

5. send the global class centers c^t to C_i

6. for $t=1,2,\cdots,T-1$ in parallel do

7. $c_i^t \leftarrow c^t$

8. $\theta_i^t \leftarrow \theta^t$

9. $h_i \leftarrow p_i^{t-1} \circ o_i^{t-1}$

10. $w_i^t \leftarrow \theta_i^t \circ h_i$

11. $d_i^t \leftarrow w_i^t$

12. PartyLocalTraining (i,t):

13. for epoch $s=1,2,\cdots,\tau_{\mathrm{h}}$ do

14. for epoch batch $b=x,\ y$ of D^i do

15. $l_{\mathrm{sup}} \leftarrow \mathrm{CrossEntropyLoss}(F_{w_i^t}(x), y)$

16. $h_i^t \leftarrow h_i^t - \eta \nabla l_{\mathrm{sup}}$

17. for epoch $i=1,2,\cdots,\tau_{\varnothing}$ do

18. for epoch batch $b=x,\ y$ of D^i do

19. $z = P_{w_i^t}(x)$

20. $z_{\mathrm{con}} = P_{w_i^t}(x)$

21. $z_{\mathrm{glob}} = P_{w_i^t}(x)$

22. $l_{\mathrm{sup}} \leftarrow \mathrm{CrossEntropyLoss}(F_{w_i^t}(x), y)$

23. $l_{\mathrm{con}} \leftarrow$

24. $-\log \dfrac{\exp(\mathrm{sim}(z, z_{\mathrm{glob}})/\tau)}{\exp(\mathrm{sim}(z, z_{\mathrm{glob}})/\tau + \exp(\mathrm{sim}(z, z_{\mathrm{con}})/\tau}$

25. $l = l_{\mathrm{sup}} + l_{\mathrm{con}}$

26. $\theta_i^t \leftarrow \theta^t - \eta \nabla l$

27. $\theta_i^{t+1}, c_i^{t+1} \leftarrow \mathrm{PartyLocalTraining}(i, t, w^t, c^t)$

28. return $\theta_i^{t+1}, c_i^{t+1}$ to server

29. $\theta^{t+1} \leftarrow \dfrac{1}{n}\sum_{k=1}^{n} \dfrac{|D^i|}{|D|}\ \theta_i^{t+1}$

30. $c^{t+1} \leftarrow \dfrac{1}{n}\sum_{k=1}^{n} \dfrac{|D^i|}{|D|}\ c_i^{t+1}$

31.return $w_1^T, w_2^T, \cdots, w_n^T$

When round $t=0$, the server initializes the model w^0, c^0 and sends them to all clients. In other rounds, the server receives a local base encoder model θ_i^t from participants, and updates them by weighted average method to obtain θ_i^{t+1}, then sends it to the participants in the next round. In addition to initialization, the communication process only transmits partial network parameters. In party-side training, the party updates the model w_i^t using local data via SGD and updates each class center c_i.

4.4 Privacy Protection Based on Gradient Leakage

Neural network models are usually trained by a hot-label (one-hot) cross-entropy loss function, which can be defined as:

$$L(x,k) = -\sum_{k=1}^{M} y_i \log\left(p_i\right) \tag{7}$$

where x is the input data, k is the corresponding ground truth label, M is the number of classes. We have $y_i=1$ when $i=k$, otherwise $y_i=0$. And $s=[s_1,s_2,\cdots]$ is the prediction score of input x through neural network, and p_i denotes the output of s_i after the Softmax activation function.

The gradient vector ∇W_L^i of the weight W_L^i connected to the i-th logit can be written as:

$$\begin{aligned}\nabla W_L^i &= \frac{\partial L}{\partial W_L^i} = \frac{\partial L_i}{\partial p_i} \cdot \frac{\partial p_i}{\partial s_i} \cdot \frac{\partial s_i}{\partial W_L^i} \\ &= -\frac{1}{p_i}\left[p_{ik} \cdot \left(p_i - y_i\right)\right] h_i \\ &= \sigma\left(s_i\right) - y_i\end{aligned} \tag{8}$$

Based on this independent of the model architecture and parameter rules, it is possible to identify the ground-truth label k of the private training data x from the shared gradient ∇W. In other words, this inference is applicable to any network in any training phase from any random initialization of the parameters[23].

Gradient-based attacks require access to the complete gradient information, especially in the last layer. An intuitive defense strategy is gradient masking, which transmits incomplete gradient information that does not affect collaborative modeling. We design a layered network structure that locally preserves the gradient information of the personalized layers. For instance, when the last layer is masked, the attacker can only infer the label from the gradient information of the inverted second layer. The gradient vector ∇W_L^i of the weight W_{L-1}^i connected to the i-th logit in the output layer can be written as:

$$\begin{aligned}\nabla W_{L-1}^{i} &= \frac{\partial L(x,k)}{\partial W_{L-1}^{i}} \\ &= \left(\sum_{j=1}^{s} \frac{\partial L(x,k)}{\partial s_j} \cdot \frac{\partial s_j}{\partial h_i} \right) \cdot \frac{\partial h_{L-1}^{i}}{\partial W_{L-1}^{i}} \\ &= \left(\sum_{j=1}^{s} \left(\sigma\left(s_j\right) - y_j \right) \cdot W_{L}^{ij} \right) \cdot h_{L-2}^{i} \end{aligned} \tag{9}$$

where W_L^{ij} denotes the weight parameter of the $L-1$ layer weight parameter associated with the hidden layer neuron h_{L-1}^{i}. The sign of ∇W_{L-1}^{i} is associated with the uncertain value of W_L^{ij}, so the gradient information and labeling relationship cannot be accurately determined by the above conclusion. The experimental validation process is described in Section 5.10.

5 Experiment Studies

To demonstrate the superiority of this work, the MpFedcon is compared with the state-of-the-art federated learning algorithms. The global FL approaches include FedAvg[4], Fedprox[20], SCAFFOLD[13], The personalized FL approaches, such as Per-FedAvg[27] uses meta-learning to learn an initial model before adapting to each task to fine-tune it. APFL[21] interpolates between local and global models, and Ditto[19] learns local models and encourages these models to be tightly coupled through global regularization. Fedper[28], Fedrep[29] are also a layered network architecture, as they learn a global representation and personalization head. However, these methods do not explore privacy protection. We use SOLO as a baseline method. Recall that the SOLO approach involves each party training a model with local data without federated learning. Further, we compare the single global model and its fine-tuned approach. To obtain the fine-tuning results, we first train the global model for the entire training cycle, and then each party fine-tunes its local training data by 10 SGD only, then calculate the final test accuracy.

5.1 Experimental Setup

Experiments are conducted over three standard datasets: CIFAR-10, CIFAR-100[43] and FEMNIST[44]. The heterogeneity of the CIFAR-10 and CIFAR-100 is controlled by assigning different class numbers S to each party. Each party is assigned the same number of training samples. For FEMNIST, the dataset is restricted to 10 handwritten letters, and samples are assigned to the parties according to the log-normal distribution[38]. There is a partition containing 150 parties, with an average of 148 samples/parties. As in the previous work[28], a 5-layer CNN model is used as the base encoder for CIFAR-10 and CIFAR-100, and a 2-layer MLP for FEMNIST. The projection head for all methods consists of a 2-layer MLP, while the output layer is a single linear layer. MpFedcon performs 10 SGD local epochs with momentum to train the local head, followed by one epoch for the base encoder layer in the case of CIFAR-10 and five epochs in all other cases. All other methods use the same number of local epochs as MpFedcon to update the base encoder layer. The accuracy is calculated by taking the average local accuracy of all users in the last 10 rounds of communication.

5.2 Accuracy Results

Table 1 lists the top-1 test accuracy of all methods. The SOLO method has better performance results since it can fit the local data preferably, as the data assigned to each party is small and biased. The data skew distribution severely impairs the performance of FedAvg. The SCAFFOLD and FedProx methods based on FedAvg perform much worse than FedAvg, so it may be difficult to find the right direction to correct data heterogeneity. Furthermore, APFL and Ditto outperform the classical FedAvg performance because the hybrid and regularization methods partly bridge the local and global model drift. Surprisingly, the fine-tuned FedAvg method performs well, probably because the fine-tuning adapts to the unique data distribution. Fedper and Fedrep methods

Table 1 The top-1 accuracy of MpFedcon and the other methods on test datasets

%

Method	CIFAR-10		CIFAR-100		FEMNST
	(100,2)*	(100,5)	(100,5)	(100,20)	(150,3)
SOLO	89.79	70.68	75.29	41.29	60.86
FedAvg	42.65	51.78	23.94	31.97	51.64
FedAvg+FT	87.65	73.68	79.34	55.44	72.41
FedProx	39.92	50.99	20.17	28.52	18.89
SCAFFOLD	37.72	47.33	20.32	22.52	16.65
APFL	83.77	72.29	78.20	55.44	70.74
Ditto	85.39	70.34	78.91	56.34	68.28
PerFedAvg	82.27	67.20	72.05	52.49	71.51
FedPer	87.13	73.84	76.00	55.68	76.91
Fedrep	87.70	73.84	79.15	56.10	78.56
MpFedcon	89.35	77.61	79.99	56.37	83.30

* means the number of clients is 100 and the number of classes is 2

based on layered networks further improve the accuracy. However, none of the above methods addresses the inconsistency between local and global optimization objectives due to data heterogeneity, which affects the final performance results. It can be observed that MpFedcon performs the best on the datasets with different degrees of heterogeneity. Subject to similar semantic interference, the MpFedcon has a 0.27% to 0.84% accuracy improvement on CIFAR-100 with hyper-classification, more than 1.65% on CIFAR-10, and more than 4.74% on FEMNIST. It shows that MpFedcon effectively improves the federated learning effect.

5.3 Effect of Data Heterogeneity

To evaluate the effect of heterogeneity, we control the degree of heterogeneity of each party by varying the number of classes S. For the CIFAR dataset, the number of training samples per party is equal to 50 000/n, where n represents the number of parites, so columns with 100 parties have 500 training samples per party. Comparatively, columns with 1 000 parties have only 50 training samples per party. As we can see from Table 1, MpFedcon always achieves the best accuracy in all cases. The advantage of MpFedcon is the introduction of class centers, which can be used as global knowledge to correct local training, and personalized classification layers further fit local data to improve the classification.

5.4 Impact of Global Communication Rounds (T)

Figure 6 shows the accuracy of each round during the training period. MpFedcon achieves the best performance at the end of training. In addition, the curves in Fig. 6 show that MpFedcon sacrifices the convergence speed in the early stages because the learning class central features affect the overall optimization direction in the early stages. The FedAvg, Fedprox, and SCAFFOLD converge slowly and fluctuate greatly with increased communication rounds. It shows that the methods of sharing the same network or modifying the gap between the local and global networks are not applicable under heterogeneous settings. Although Fedper and Fedrep, based on simple layered networks, learn quickly in the early stage, MpFedcon performs better in the later stages. In other words, a better class centered representation gives the classifier better classification ability at a later stage.

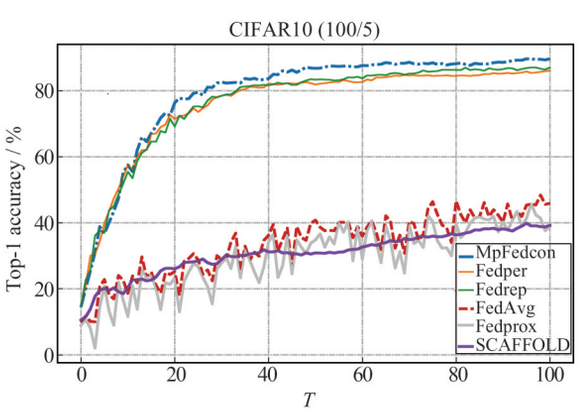

Fig. 6 Top-1 test accuracy with different number of communication rounds (T)

5.5 Influence of Local Epoch Number (E)

We study the influence of local epoch numbers on the accuracy of the final model. Figure 7 shows the effect on accuracy and convergence speed during training. The accuracy and convergence speed are reduced when the number of local epochs is 1, especially FedAvg. It can be observed in Fig. 8 that when the number of local epochs $E = 10$, most methods have the highest accuracy and faster convergence. This is because when E is small,

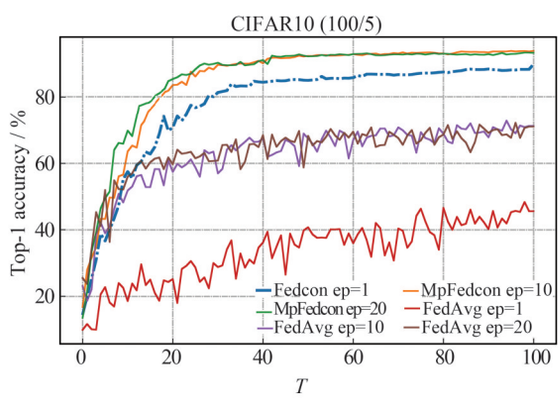

Fig. 7 Top-1 test accuracy curves of different local epoch numbers

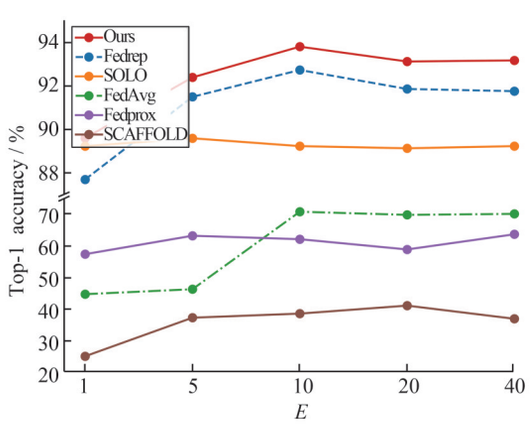

Fig. 8 Top-1 test accuracy line chart of local epoch number (E) of different algorithms

the local networks cannot be fully trained and converge slowly. However, the improvement of accuracy and convergence speed will be slight when $E>10$, and there may be overfitting for local training of skewed data, which leads to a decrease in the accuracy of the global model.

5.6 Scalability

To demonstrate the scalability of MpFedcon, we use different numbers of parties to participate in the training on the CIFAR-10 dataset. Specifically, we try two settings: 1) the dataset is divided into 50 parties and 5 parties per round are randomly selected; 2) the dataset is divided into 100 parties and 10 parties in each round of federated training are randomly selected. The results are shown in Table 2. For MpFedcon, the results are shown for $\mu=0.5$, 5, 10, which best outperforms Fedrep with over 2% accuracy at 200 rounds with 50 parties and 5% accuracy at 200 rounds with 100 parties. Partial party participation means that the active data is only a subset of all training data, which leads to unstable training and slower convergence. MpFedcon consistently achieves the best performance with the participation of different parts in Table 2, which shows that the performance of MpFedcon will not be affected by the increases in the number of parties.

Table 2 Top-1 test accuracy with varying number of parties (*m*) and communication rounds (*T*) on CIFAR-10 (heterogeneity: 100/5) %

Method	*m*=50		*m*=100	
	T=100	*T*=200	*T*=100	*T*=200
FedAvg	48.62	55.15	51.30	62.92
FedProx	63.73	65.07	69.07	73.38
Fedrep	72.98	74.21	75.68	77.64
MpFedcon(μ=0.5)	71.50	76.38	70.38	77.28
MpFedcon(μ=5)	72.65	77.22	75.68	81.14
MpFedcon(μ=10)	74.69	79.31	77.61	83.14

5.7 Effect of Coefficient in the Loss Function (μ)

In this work, we use the coefficient μ to adjust the weights of the classes' centers feature learning and classifier learning during training. Different coefficient μ of experiments are explored on CIFFAR-10. Specifically, μ is a hyperparameter used to weigh the class center's optimization direction against its dataset's optimization direction. As shown in Table 2, MpFedcon achieves the best results when μ=10. A smaller coefficient μ increases the fitting effect of the personalization layer on a small amount of local data, thus improving the model accuracy. While a larger μ slows down the convergence in the short term, it improves the overall classification effect in subsequent exchanges.

5.8 Communication Efficiency

The communication overhead of federated learning is mainly caused by the transfer of data (e.g., models, parameters) between the party and the central server. Many current studies focus on studying the reduction of one aspect, such as reducing the number of communications without caring about the cost of a single transmission. We believe that a more credible metric for judging the communication cost is the total amount of communication data at convergence. It can be expressed as:

$$\text{Traffic} = \text{round} * \text{traffic}_1 \qquad (10)$$

where Traffic is the total communication volume, rounds is the number of communications, and traffic_1 denotes a single traffic volume. For a fair comparison, each algorithm uses the same network structure with single traffic of 1.2 and 2.2 MB for CIFAR-10 and CIFAR-100, respectively.

FedAvg reduces the number of communications by increasing the number of local updates. FedAvg algorithm converges under both iid data and non-iid data. However, the convergence speed of FedAvg is limited by the distribution state of the dataset. As shown in Table 3, the most representative algorithms are compared in heterogeneous environments to obtain the same accuracy. FedAvg sacrifices the communication cost to improve the model's accuracy. Fedprox and Fedrep have the same single-transfer cost as FedAvg, benefiting from

Table 3 Accuracy with 50 parties and 100 parties (sample fraction=0.1) on CIFAR-10 and CIFAR-100 (heterogeneity: 100/5)

Dataset	Method	Round	Speedup	Traffic/MB
CIFAR-10	FedAvg	200	1	240
	FedProx	116	1.72	139.2
	Fedrep	58	3.40	68.3
	MpFecon(ours)	54	3.70	64.1
CIFAR-100	FedAvg	200	1	440
	FedProx	179	1.1	393.8
	Fedrep	4	50.0	8.8
	MpFecon(ours)	7	28.5	15.4

the convergence speed and smaller total communication cost. Especially Fedrep's personalized model dramatically improves the convergence speed and has the smallest communication cost. Compared with FedAvg and Fedrep, MpFedcon adds a smaller amount of additional class center features to be transmitted. But with the increase in data volume and communication rounds, MpFedcon has a greater advantage in terms of computational cost. The contrastive loss term can effectively improve the accuracy without reducing the overall convergence speed.

5.9 Effectiveness of MpFedcon

For demonstration purposes, we use the SOLO and the most classical FedAvg method to evaluate the effectiveness of MpFedcon. We take the SOLO method of each party as the test baseline, then Fig. 9 visualizes the improvement of each party after passing the MpFedcon and FedAvg. As shown in Fig. 9, MpFedcon effectively improves precision for more than 70% participating parties in a highly heterogeneous setting. However, the classic Fedavg has almost no accuracy improvement for the participating parties. The classical FedAvg approach almost fails in the case of data heterogeneity.

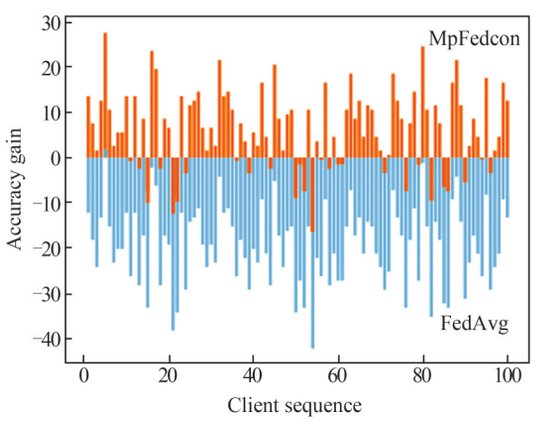

Fig. 9 Effectiveness of precision improvement of MpFedcon and FedAvg for 100 party segments involved in training

5.10 Gradient Leakage-Based Privacy Defense

For a fair comparison, experiments are shown on the CIFAR-10 and CIFAR-100 datasets for the classification tasks according to the settings in iDLG[22]. LeNet is initialized with random for all experiments, and we use L-BFGS[45] with a learning rate of 1 as the optimizer.

The gradient attack is visualized by the same conditions for the same random image as in Fig. 10. The curve represents the MSE between the generated image and the real image. Then we visualize the final image generated by each method. MpFedcon masks the gradient information of the classification layer, and the attacker cannot accurately know the number of parties' personalized layers and sensitive information. To effectively test the possible cases of gradient attack, the experiments verify the effect of gradient attack by setting cl/at, where cl is the number of FC layers on the party side, at is the number of FC layers on the attacker side. Even with only 1 FC layer, the attacker still fails to identify effectively after many iterations. However, the DLG and iDLG methods accurately restore the image after 50 rounds of iterations.

In addition, it can be seen from Fig. 10 that the more FC layers the party masks, the bigger the error caused and the more difficult it is to be attacked. Table 4 shows the traditional defenses based on Gaussian noise and Laplace noise with a large variance of 10^{-2} effectively defend against noise defense, but both severely degrade the accuracy[22]. The results show that MpFedcon effectively resists the privacy leakage problem of gradient-based attacks while ensuring the model's accuracy.

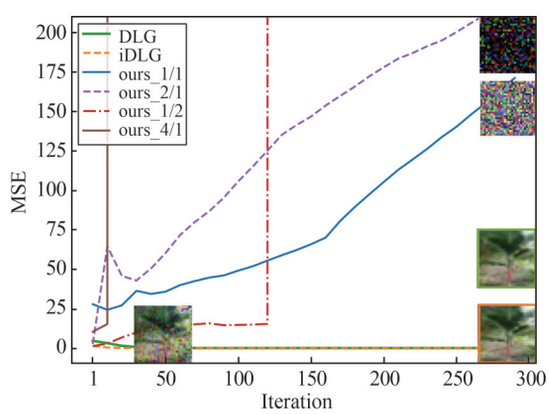

Fig. 10 The effectiveness of various defense strategies

Table 4 Testing datasets performance

%

	Original	G - 10^{-4}	G - 10^{-2}	L - 10^{-4}	L - 10^{-2}	Ours
Accuracy	76.3	75.6	45.3	75.6	46.2	**76.1**
Defendability	—	×	√	×	√	√

Note: G means Gaussian noise, and L means Laplace noise

6 Conclusion

Non-iid is a significant obstacle to the availability of federated learning. To improve the performance of federated learning models on non-iid datasets, we pro-

pose a new MpFedcon algorithm with resistance to label leakage. Specifically, MpFedcon uses all party's data to learn a global representation and corrects the local optimization direction to be consistent with the global distribution by model-contrastive loss with the class center. Utilizing the computing resources of parties to conduct numerous local updates can further fit the local data distribution while retaining sensitive information to prevent label disclosure. Extensive experiments on various image classification datasets demonstrate the advantage of MpFedcon on non-iid data. As MpFedcon does not require the inputs to be images, it is potentially applied to non-vision problems.

References

[1] Weber P A, Zhang N, Wu H M. A comparative analysis of personal data protection regulations between the EU and China [J]. *Electronic Commerce Research*, 2020, **20**(3): 565-587.

[2] Guo P F, Wang P Y, Zhou J Y, *et al*. Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning [C]//2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*). New York: IEEE, 2021: 2423-2432.

[3] Kairouz P, McMahan H B, Avent B, *et al*. Advances and open problems in federated learning [J]. *Foundations and Trends® in Machine Learning*, 2021, **14**(1/2): 1-210.

[4] McMahan H B, Moore E, Ramage D, *et al*. Communication-efficient learning of deep networks from decentralized data [EB/OL]. [2022-09-17]. *https://arxiv.org/abs/*1602.05629.

[5] Mothukuri V. A survey on security and privacy of federated learning [J]. *Future Generation Computer Systems*, 2021, **115**: 619-640.

[6] Wang X F, Wang C Y, Li X H, *et al*. Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching [J]. *IEEE Internet of Things Journal*, 2020, **7**(10): 9441-9455.

[7] Samarakoon S, Bennis M, Saad W, *et al*. Distributed federated learning for ultra-reliable low-latency vehicular communications [J]. *IEEE Transactions on Communications*, 2020, **68**(2): 1146-1159.

[8] Begum A M, Mondal M R H, Podder P, *et al*. Detecting spinal abnormalities using multilayer perceptron algorithm [C]// *Innovations in Bio-Inspired Computing and Applications*. Cham: Springer International Publishing, 2022: 654-664.

[9] Dong J H, Cong Y, Sun G, *et al*. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation [C]// 2020 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*). New York: IEEE, 2020: 4022-4031.

[10] Yang Q, Zhang J Y, Hao W T, *et al*. FLOP: Federated learning on medical datasets using partial networks [C]// *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. New York: ACM, 2021: 3845-3853.

[11] Ramaswamy S, Mathews R, Rao K, *et al*. Federated learning for emoji prediction in a mobile keyboard [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*1906.04329.

[12] Duan M M, Liu D, Chen X Z, *et al*. Self-balancing federated learning with global imbalanced data in mobile systems [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2021, **32**(1): 59-71.

[13] Karimireddy S P, Kale S, Mohri M, *et al*. SCAFFOLD: Stochastic controlled averaging for on-device federated learning [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*1910.06378.

[14] Khaled A, Mishchenko K, Richtárik P. Tighter theory for local SGD on identical and heterogeneous data [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*1909.04746.

[15] Jiang M R, Wang Z R, Dou Q. HarmoFL: Harmonizing local and global drifts in federated learning on heterogeneous medical images [J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, **36**(1): 1087-1095.

[16] Li T, Sahu A K, Zaheer M, *et al*. Federated optimization in heterogeneous networks [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*1812.06127.

[17] Hsieh K, Phanishayee A, Mutlu O, *et al*. The non-IID data quagmire of decentralized machine learning [C]// *Proceedings of the 37th International Conference on Machine Learning*. New York: ACM, 2020: 4387-4398.

[18] Wang J Y, Liu Q H, Liang H, *et al*. Tackling the objective inconsistency problem in heterogeneous federated optimization [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*2007.07481.

[19] Li T, Hu S Y, Beirami A, *et al*. Ditto: Fair and robust federated learning through personalization [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*2012.04221.

[20] Li T, Sahu A K, Zaheer M, *et al*. Federated optimization in heterogeneous networks [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*1812.06127.

[21] Deng Y Y, Kamani M M, Mahdavi M. Adaptive personalized federated learning [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*2003.13461.

[22] Zhu L, Liu Z, Han S. Deep leakage from gradients [EB/OL]. [2022-09-23]. *https://arxiv.org/pdf/*1906.08935.

[23] Zhao B, Mopuri K R, Bilen H. iDLG: Improved deep leakage from gradients [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*2001.02610.

[24] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning [C]// *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*. New York: ACM, 2017: 4080-4090.

[25] Chen X L, He K M. Exploring simple Siamese representation learning [C]// 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*). New York: IEEE, 2021: 15745-15753.

[26] Li Q B, He B S, Song D. Model-contrastive federated learning [C]//2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*). New York: IEEE, 2021: 10708-10717.

[27] Fallah A, Mokhtari A, Ozdaglar A. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach [C]// *Proceedings of the* 34*th International Conference on Neural Information Processing Systems*. New York: ACM, 2020: 3557-3568.

[28] Arivazhagan M G, Aggarwal V, Singh A K, *et al*. Federated learning with personalization layers [EB/OL]. [2022-09-30]. *https://arxiv.org/abs/*1912.00818.

[29] Collins L, Hassani H, Mokhtari A, *et al*. Exploiting shared representations for personalized federated learning [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*2102.07078.

[30] He K M, Fan H Q, Wu Y X, *et al*. Momentum contrast for unsupervised visual representation learning [C]// 2020 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*). New York: IEEE, 2020: 9726-9735.

[31] Chen T, Kornblith S, Norouzi M, *et al*. A simple framework for contrastive learning of visual representations [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*2002.05709.

[32] Khosla P, Teterwak P, Wang C, *et al*. S3 upervised contrastive learning [EB/OL]. [2022-09-23]. *https://arxiv. org/abs/* 2004.11362.

[33] van Berlo B, Saeed A, Ozcelebi T. Towards federated unsupervised representation learning [C]// *Proceedings of the Third ACM International Workshop on Edge Systems*, *Analytics and Networking*. New York: ACM, 2020: 31-36.

Zhang F D, Kuang K, You Z Y, *et al*. Federated unsupervised representation learning [EB/OL]. [2022-09-23]. *https://arxiv. org/abs/*2010.08982.

[34] Wang P, Han K, Wei X S, *et al*. Contrastive learning based hybrid networks for long-tailed image classification [C]// 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*). New York: IEEE, 2021: 943-952.

[35] Wang W, Zhou T, Yu F, *et al*. Exploring cross-image pixel contrast for semantic segmentation [C]//2021 *IEEE/CVF International Conference on Com-puter Vision* (*ICCV*). New York: IEEE, 2021: 7283-7293.

[36] Song G C, Chai W. Collaborative learning for deep neural networks [C]// *Proceedings of the* 32*nd International Conference on Neural Information Processing Systems*. New York: ACM, 2018: 1837-1846.

[37] Wainakh A, Ventola F, Müßig T, *et al*. User-level label leakage from gradients in federated learning [J]. *Proceedings on Privacy Enhancing Technologies*, 2022, **2022**(2): 227-244.

[38] Li T, Sahu A K, Zaheer M, *et al*. FedDANE: A federated Newton-type method [C]//2019 53*rd Asilomar Conference on Signals*, *Systems*, *and Computers*. New York: IEEE, 2019: 1227-1231.

[39] Zhao Y, Li M, Lai L Z, *et al*. Federated learning with non-iid data[EB/OL].[2022-09-23]. *https://arxiv.org/abs/*1806.00582.

[40] Yu F X, Rawat A S, Menon A K, *et al*. Federated learning with only positive labels [C]// *Proceedings of the* 37*th International Conference on Machine Learning*. New York: ACM, 2020: 10946-10956.

[41] van der Maaten L, Hinton G. Visualizing data using t-SNE [J]. *Journal of Machine Learning Research*, 2008, **9**: 2579-2625.

[42] Oord A V D, Li Y Z, Vinyals O. Representation learning with contrastive predictive coding [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*1807.03748.

[43] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[EB/OL].[2022-09-23]. *https://www.semanticscholar. org/paper/Learning-Multiple-Layers-of-Features-from-Tiny-Krizhevsky/*5*d*90*f*06*bb*70*a*0*a*3*dced*62413346235*c*02*b*1*aa*086.

[44] Caldas S, Duddu S M K, Wu P, *et al*. LEAF: A benchmark for federated settings [EB/OL]. [2022-09-23]. *https://arxiv.org/abs/*1812.01097.

[45] Liu D C, Nocedal J. On the limited memory BFGS method for large scale optimization [J]. *Mathematical Programming*, 1989, **45**(1): 503-528.

□