FedPG: A Privacy-friendly and Universal Method for Solving non-IID Data in Federated Learning

Baolu Xue[1*], Jiale Zhang[2,3], Bing Chen[1,4*], Weizhi Meng[5,6]

[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China.
[2]College of Information Engineering, Yangzhou University, Yangzhou, China.
[3]School of Mathematics, Physics and Computing, University of Southern Queensland, Toowoomba, Australia.
[4]Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China.
[5]Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark.
[6]School of Computing and Communications, Lancaster University, Lancaster, UK.

*Corresponding author(s). E-mail(s): xbl236@nuaa.com; cb_china@nuaa.edu.cn;
Contributing authors: jialezhang@yzu.edu.cn; weme@dtu.dk;

Abstract

Federated Learning (FL) is a privacy-preserving distributed learning framework which could harness the potential of decentralized multimedia data. However, a significant hurdle lies in the non-uniform distribution of data among clients, leading to slow convergence and subpar accuracy in the global model. Although several approaches have been proposed to address this challenge, two key limitations remain. First, these methods frequently require access to information about local data or even the raw data, which raises significant privacy concerns for clients. Second, these methods struggle to perform well in a common non-IID scenario: class missingness, and they often fail to fully resolve the issue of client drift. In response, in this paper, we propose a privacy-friendly and universal method FedPG to solve non-IID data in FL. The core idea behind FedPG is to leverage homogeneous virtual data to alleviate both data heterogeneity and client drift.

Specifically, FedPG introduces a novel image generation method based on prototype loss, which does not require any additional privacy-sensitive information. This approach generates synthetic datasets aligned with the global distribution to effectively assist local training. Besides, we also design a local training method that is suitable for scenarios involving class missingness, enabling both feature adaptation and classifier de-biasing. The comprehensive experiments demonstrate the efficacy of our FedPG framework. In the majority of cases, FedPG not only achieves superior accuracy but also exhibits accelerated convergence rates compared to alternative approaches.

Keywords: Federated Learning, data heterogeneity, prototype, domain adaptation

1 Introduction

Recently, widespread use of multi-media sensors has led to the generation and storage of large amounts of decentralized multi-media data. Utilizing this data to drive Artificial Intelligence (AI) technology holds promise for accelerating intelligent system development and improving service quality [1, 2]. However, sharing raw data raises serious privacy concerns and incurs substantial communication costs. Fortunately, Federated Learning (FL)[3, 4] has been proposed to harness the potential of such distributed multimedia data and facilitate the development of high-quality AI models. Computer vision tasks, in particular, have been extensively studied and applied in FL[5]. In this context, clients train visual models on-device, ensuring that data remains on the local device. This preserves privacy and reduces associated data transmission and storage costs. However, due to the independent generation of data on distributed multi-media sensors, the data across client devices exhibit a non-independent and identically distributed (non-IID) nature, commonly referred to as data heterogeneity[6, 7]. Data heterogeneity is one of the significant challenges of FL. It has been identified leading to oscillatory and slow convergence and contributing to suboptimal model performance[7–9]. Moreover, the performance of FL tends to degrade when facing class-missing data, an extreme yet common form of data heterogeneity [10]. For instance, a hospital specializing in heart diseases may entirely lack samples related to respiratory conditions.

Data heterogeneity in FL leads to client-drift, where local models converge towards their individual optima due to non-IID data, causing the global model to deviate from the global optimum[11]. To address client-drift issue, a natural approach is to improve the generalization ability of local models, thereby enhancing the overall performance of the global model. And numerous related works have been developed based on this idea, which can be categorized into two types: **One** aims to enhance the similarity among local models[11–17]. Two prevalent methods for achieving this are feature alignment and classifier de-biasing, which target the feature extractor and classifier components, respectively. However, these methods prove ineffective when dealing with class-missing local data, as they rely on the presence of all classes to guide local optimization towards the global optimum. Moreover, as noted in prior works[6, 9], it's challenging for a single

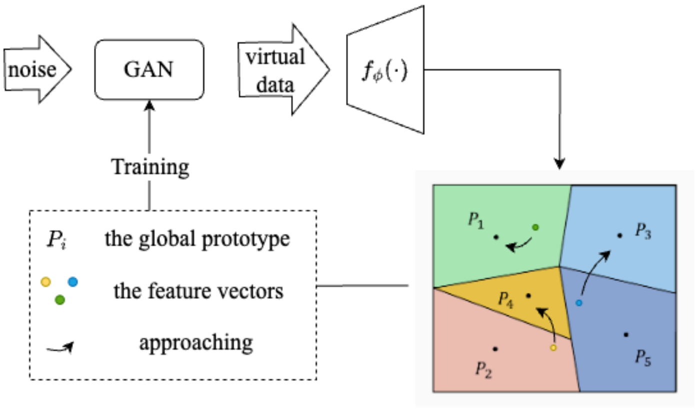


Fig. 1: An illustration of GAN training on the server side in our FedPG. Given a label, the GAN generates virtual images, which are passed through the global model to obtain their feature representations. The GAN is optimized to align these features with the corresponding global prototypes, ensuring the synthetic datasets are consistent with the system's data distribution.

learning paradigm to address client drift problem across varying data distributions. **Another** focuses on alleviating the heterogeneity among local data. Compared to the previous category, these methods directly address the root cause of client drift, i.e. non-IID data. These methods involve the introduction of public data or information[7, 18–20], and clustering clients based on their local data[21–24]. However, these approaches are constrained by privacy concerns, as they often require sensitive information about local data, such as class-wise sample counts or feature distributions. Some approaches even need sharing raw data, which fundamentally contradicts the core principles of FL. Moreover, public datasets may either be unavailable or unsuitable for the specific target tasks.

To address the limitations of existing works, in this paper, we propose an effective, universal and privacy-friendly method FedPG for computer vision tasks. In FedPG, the server trains a generative model based on **P**rototype[25] and **G**enerative learning[26]. Specifically, FedPG develops a generative model capable of generating virtual images based on a given target label. As shown in Fig.1, The feature representations of virtual images, extracted by the global model, are then clustered around the corresponding global prototypes. This process ensures that the synthetic datasets are well-aligned with the system's dataset. Besides, clients share only their local prototypes in FedPG, which poses no risk to privacy[27–29]. By sharing the virtual data among clients, homogeneity is introduced into their local datasets[20]. We also propose a local training method to correct the drift of clients' models more thoroughly. It aligns the features of real and virtual images from the same class to achieve domain adaptation[30], while incorporating virtual data into classification to de-bias the classifier. This approach does not assume locally complete class distributions, as the synthetic dataset aids in generalizing local models even in class-missing scenarios.

Our FedPG reaps multifold advantages: (1) By introducing homogeneous virtual dataset, FedPG is able to improve the generalization capability of local models. It could enhance the performance of global model in turn. (2) It harmonizes with the objective of safeguarding privacy in FL. On one hand, it doesn't depend on sharing raw data or statistical information, consistent with the principle of FL. On the other hand, updating local prototypes is privacy-friendly since the computation process is irreversible[27]. (3) Compared to some prior work, our method is more universal. FedPG uses virtual prototypes to overcome the miss of global prototypes of certain classes. So there are no extra constraints for clients selection strategy. Besides, with the use of virtual datasets, it can function effectively even in the presence of class-missing data. Our comprehensive experiments have validated the efficacy of FedPG from multiple perspectives, demonstrating a performance improvement both on convergence rate and test accuracy compared to the baseline. We evaluate the performance on three image classification datasets, including CIFAR-10, CIFAR-100[31], and FEMNIST[32], across three data partitioning methods. In most cases, FedPG achieves state-of-the-art performance. For example, on CIFAR-10 dataset with four non-IID settings, our method achieves a performance improvement of 3.59% to 5.03% over FedAvg, while also achieving convergence twice as fast as FedAvg.

The main contributions of this paper are as follows.

- We propose a universal and privacy-friendly algorithm, FedPG, which leverages prototype loss to train a generative model capable of producing public datasets to introduce homogeneity to clients.
- We propose a local training strategy that addresses both feature adaptation and classifier de-biasing, with the goal of mitigating client drift, including scenarios involving class-missing local data.
- We conduct comprehensive experimental evaluations across three benchmark datasets, considering three distinct scenarios of data heterogeneity, and demonstrate the effectiveness of our FedPG method.

The rest of this paper is structured as follows. Section 2 introduces the data heterogeneity issue in FL and discusses two mainstream approaches based on the FedAvg framework. Section 3 presents the system definition and outlines our motivation. Section 4 provides the details of our proposed method, FedPG. The experimental setup, results, and analysis are presented in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

In the section, we introduce one of the most traditional methods in FL, FedAvg. There are various methods to tackle non-IID issue based on FedAvg framework, and we focus on the algorithms enhancing local model generalization. We divide the related work into two categories and compare our method with these baselines.

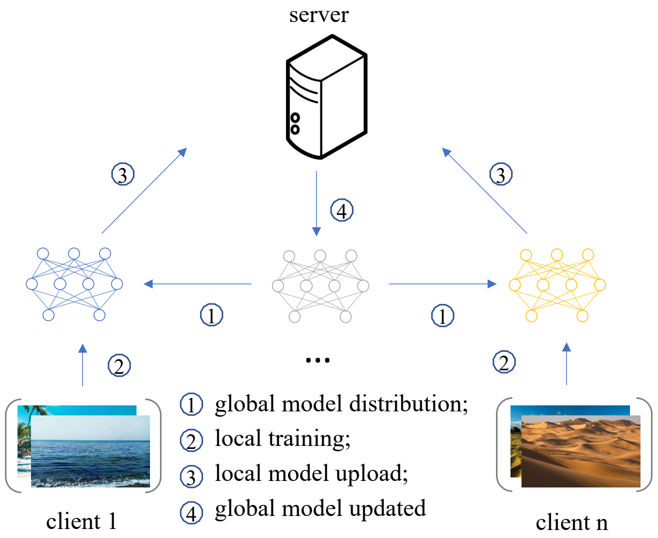


Fig. 2: The framework of algorithm FedAvg.In FedAvg, each client trains a local model on its own data and periodically shares model updates with central server. The server then averages these updates to obtain a global model and sent it back to clients. The process repeats iteratively until system convergence.

2.1 Federated Learning

FL is a rapidly evolving research domain of distributed learning. One of the most popular algorithms is FedAvg[33], whose progress is illustrated in Fig.2. At the beginning, server initializes global model and then distributes it to a fraction of clients selected randomly. When receive the global model, clients train it with local data and send local models to server. As soon as a sufficient quantity of local models is received, server aggregates them to update global model. In the next round, server continues to distribute the updated global model to the selected clients. The above steps are repeated until the model has converged. FL also presents several open challenges [6], such as the issue of non-IID data. Since clients generate local data independently, the data distributions across clients are heterogeneous, which leads to subpar performance in FL, including slow convergence and suboptimal model quality.

Based on framework of FedAvg, there are two mainstream classes of methods to address the issue[17]. One focuses on optimizing local training. These methods encompass incorporating the global context into local updates to generalize local models. Another aims to design aggregation strategies to smoothly transfer knowledge from local models to the global one[17, 34]. Compared to the former, the latter can't address inherent heterogeneity among local models arising from local training. Moreover, the advanced aggregation strategies can be used besides local training optimization in most cases. So in the work, we focus on the methods enhancing local model generalization and categorize the related work into two main types.

2.2 Enhancing Similarity of models

Data heterogeneity presents itself as heterogeneity among local models' parameters. Many approaches aim to mitigate this client drift by constraining the update directions to make local models more similar. Existing works achieve similarity among local models from three perspectives:

(i) The Whole Model: Some works align local models to the global model by considering model parameters, logits, or gradients [8, 11, 12, 35]. For example, in FedNTD[35], clients constraint the predictions of the same sample generated by both the local and global models. However, this approach may overlook the inner significance of model heterogeneity, potentially erasing the underlying knowledge of local models[17].

(ii) Feature Extractor: Feature alignment, as proposed in contrastive learning[19, 36, 37], aims to reduce the distance between features from the same class. In FL, it is used to learn generalized feature representations across local models. For example, in MOON[37], clients input their local data to both global and local models, constraining the distance between the output feature vectors. However, the significant bias of classifiers still hampering performance[16]. What's more, most feature alignment methods are ineffective when dealing with class-imbalanced data. For MOON, if a client has no sample for a certain class, it becomes impossible to correct the features of that class's data.

(iii) Classifier: A classifier is a component of a classification model that maps feature vectors to predictions. Ideally, the local model performs equally well across all classes present in system dataset. But due to non-IID data, classifiers may produce highly imbalanced predictions. Some works focus on classifier tuning[16, 17, 38]. The work[16] experimentally highlights that "the devil is in the classifier" and corrects the classifier using virtual representations. However, obtaining representations for classifier tuning is constrained by privacy concerns. Moreover, it is challenging to determine how the extractor will map data from locally missing class to features, making it even more difficult for the classifier to distinguish these features.

2.3 Alleviating Heterogeneity of data

Structuring more homogeneous local data is a fundamental approach to address data heterogeneity. While these methods can improve the performance of federated learning, they often introduce impractical constraints or privacy risks. There are two common categories methods:

(i) Formulating a client selection strategy[21–24] involves sampling or clustering clients based on the distributions of their local data. It could reduce the degree of heterogeneity in the training data. However, well-conceived strategies may face challenges due to varying device availability in real-world applications. Additionally, they require clients to share statistical information about local data, which is not suitable for privacy-sensitive scenarios.

(ii) Another approach entails the incorporation of additional data[7, 18–20, 39, 40]. This method shares part of the local data or a public dataset among clients to create more balanced local data. Nevertheless, sharing raw data fundamentally contradicts the core principles of FL, and additional public datasets may not be available for many applications. In consideration of this, some algorithms utilize synthetic datasets. For example, in VHL[19], the server generates a synthetic dataset using a Gaussian distribution or a Style-GAN model[41] in the initial stage. However, with the training of the system models, the fixed synthetic dataset may no longer be helpful for local training[19].

In the conclusion, two limitations are identified in the related work: the introduction of privacy concerns and the lack of robustness to class-missingness. In response, we propose a privacy-friendly and universal method, FedPG. It relies solely on local prototypes, i.e., the averages of features, to construct virtual datasets. This approach poses no risk to client privacy[27–29]. Then, it incorporates virtual dataset into local training to facilitate feature alignment and mitigate classifier bias, even in the presence of class-missing data. Hence, our approach leverages a virtual dataset to simultaneously alleviate data heterogeneity and enhance model similarity. VHL is the algorithm most closely related to our work. However, our method differs in two key aspects. First, it employs a GAN trained with a prototype loss, as opposed to being randomly initialized, to generate synthetic datasets. Second, it periodically regenerates synthetic datasets to align with the evolving training process. Our ablation experiments demonstrate the efficacy of these two design choices.

3 Notations and Preliminaries

3.1 System

This work considers training a single generalized global model for image classification tasks in FL with non-IID data. Assume there are K clients and one server in the FL system. Each client(1,...,K) has a local dataset $D_k = \{(x_i, y_i)\}_{i=1}^{|D_k|}$, where x_i and y_i indicate the ith image and its label on client k respectively. The total dataset D of whole users in FL is defined as $D = \cup_{k\in[K]} D_k$ and assume D contains C classes indexed by $[C]$. For client k, let $D_k^c = \{(x_i, y_i) \in D_k | y = c\}$ be the set of local samples with ground-truth label c. The image classification model learned in the system is represented as M, and the local model of client k is M_k. As an essential component of M, feature extractor $f_\phi : \mathcal{X} \to \mathcal{Z}$ is a mapping function from input space $\mathcal{X}$ to feature space $\mathcal{Z}$, which maps the input image x into a feature vector $z = f_\phi(x)$. And the remaining parts of M are called classifier, denote by $g_\varphi : \mathcal{Z} \to R^C$, responsible for mapping the feature vector z into a probability distribution as the predictions for input x. In the context, the parameter of the classification model is represented as $M = \{\phi, \varphi\}$.

In FL, there is round-by-round communication between clients and server. At round r, the server selects a subset of clients $k \in S^r$ to participate and distributes the global model M^{r-1} to them. Upon receiving M^{r-1}, each client locally updates it to M_k^r with the objective defined as:

$$L_k = \min_{M_k^r = \{\phi_k^r, \varphi_k^r\}} E_{(x_i, y_i)\sim D_k} \left[\mathcal{L}(M_k^r; M_k^{r-1}, x_i, y_i)\right], \tag{1}$$

where $\mathcal{L}$ indicates the loss function of local training. Take note that $\mathcal{L}$ depends on the algorithm. For example, to limit the separation between M and M_k, FedProx[8] employs the cross entropy loss with their l_2 distance; FedNTD[35] introduces a knowledge distillation loss term to preserve global knowledge during local training; And FedGen[17] uses knowledge about the distribution of global features to guide local optimizing.

In the end of round r, the server receives the optimized parameters from the selected clients S^r and updates the global model by aggregating these parameters as follows.

$$M^{r+1} = \sum_{k \in [K]} p_k M_k^r, \; where \; p_k = \frac{|D_k|}{|D|} \tag{2}$$

In general FL system, the overall objective is to learn a converged and generalized model by leveraging the local data and computational power of distributed devices. Thus, a global model that can reduce the global loss $\mathcal{L}$ over the whole dataset is required.

$$\min_M \mathcal{L}(M, D) = \frac{\sum_{k=1}^{K} \sum_{i=1}^{|D_k|} \mathcal{L}(M, x_i, y_i)}{|D|} \tag{3}$$

3.2 Motivation

When faces to data heterogeneity, the performance of the global model is oscillatory and subpar [7, 9, 11] in FL. Due to non-IID data, there is an inconsistency between the local and global tasks. In the context, local models gravitate towards their individual optima (diverging due to non-IID data) during local training, causing the global model to deviate from the global optimum[11]. Several attempts have been made to address the issue by enhancing the local model's ability to generalize, thereby improving the performance of the global model. In this section, we empirically demonstrate the existence of great bias in both extractor and classifier that would arise from data heterogeneity. Furthermore, we analyze the drawbacks of existing mitigation methods and introduce the idea of our work.

We execute FedAvg on the CIFAR-10 dataset and partition the training data among 10 clients using LDA with $\alpha = 0.1$. At the end of the 300th round, we randomly select two clients, referred to as Client 1 and Client 2, and evaluate their local models. Next, we apply T-SNE [42] to reduce the dimensionality of these feature vectors $f_\phi(x)$ to 2. The resulting visualizations are shown in Fig. 3. In the left subfigure, markers are colored based on their ground-truth labels y, while in the middle subfigure, they are colored based on the predictions $g_\varphi(f_\phi(x))$. The right subfigure illustrates the distributions of local data for Client 1 and Client 2.

Based on Fig. 3, we can observe that:

(1) There are universal feature drift among clients' local models. In the left subfigure, features generated by single model are clustered, whereas features generated by different models remain distinct. This observation highlights that in FedAvg, different local models map identical or similar inputs to divergent feature spaces. This feature drift not only weakens generalization capabilities of local models but also introduces parameters heterogeneity among them, further compromising the performance of global models.

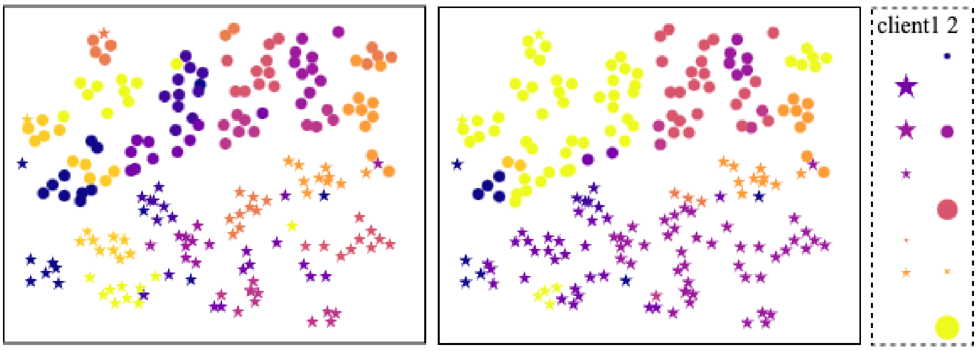


Fig. 3: The t-SNE visualization of features for two clients using FedAvg training on CIFAR-10 dataset at the 300th communication round, where distinct colors signify distinct data classes and distinct shapes signify distinct clients. The figure illustrates the significant bias in both the extractor and the classifier.

(2) Local model classifiers exhibit significant bias, with predictions strongly correlated to the underlying data distribution. As shown in the middle subfigure, even when the test data is balanced across classes, the classifier predominantly predicts feature vectors as belonging to the inner classes (i.e., classes that are prominent in the local datasets). This highlights that local models focus primarily on their immediate tasks, neglecting globally task-relevant data that is either rare or absent in their local datasets during the local updating process.

Both of these issues, feature drift and classifiers bias, occur simultaneously in setting of data heterogeneity. However, existing works[16, 17, 19, 36–38] typically only consider one of the two issues, which may be ineffective because it's always impeded by the other issue. Furthermore, as discussed in 2, the related work reveals two key limitations: the potential introduction of privacy risks and insufficient robustness in handling class-missingness scenarios. As a result, we would like to tackle both issues concurrently in order to better mitigate the non-IID issue in FL. To achieve this goal, we introduce external homogeneous datasets to local to assist local training. Next, we provide an introduction to our proposed method.

4 FedPG: Tackling non-IID in Heterogeneous Federated Learning via Prototype Learning and Generative Learning

In this section, we elaborate our proposed method FedPG. And we also analyze its effectiveness in improving model generalization and mitigate data heterogeneity.

4.1 Prototype Calculation

The mean vector of the instances belonging to the same class is referred to as the prototype[25], which enjoys the benefits of exemplar-driven nature and simpler inductive bias. In our framework, prototypes are utilized to reflect the global features of system dataset. Clients are required to compute and upload their local prototypes after local training. Subsequently, at the server side, the process commences with aggregating these local prototypes to generate the global prototypes.

Specifically, for calculating local prototypes, clients randomly select m samples from each label, where m represents a hyperparameter. Then clients input these samples into local models to obtain feature representations, For client k, the set of local prototypes $\mathcal{P}_k$ is calculated by:

$$\mathcal{P}_k = \{P_{k,c}\} = \{\frac{1}{m}\sum_{i=0}^{m} f_\phi(x_i)|x_i \in D_k^c, c \in [C]\} \tag{4}$$

We define $P_{k,c}$ as the prototype for class c on client k. It's important to note that due to data heterogeneity, the number of classes available on a client may be fewer than the total classes number C in the system dataset, meaning the size of the set $\mathcal{P}$ is less than or equal to C.

In each round, the server performs a simple aggregate to derive the set of global prototypes $\mathcal{P}$ after collecting local prototype sets. The calculation formula is as follows.

$$\mathcal{P} = \{P_c\} = \{\frac{1}{n_c}\sum_{k=0}^{|S|} P_{k,c}|c \in [C]\} \tag{5}$$

Where P_c represents the global prototype of category c. S means the set of joined clients in this round and its size is $|S|$. Besides, We define n_c as the number of prototypes belonging to category c among $\mathcal{P}_k, k \in S$.

The set of global prototypes $\mathcal{P}$ is subsequently used to guide the training of the generative model. However, since $|\mathcal{P}_k| \leq C$, $|\mathcal{P}| \leq C$. It implies that the set $\mathcal{P}$ may not include prototypes for all classes in system, which is disastrous for its subsequent utilization. To address this challenge, we utilize virtual prototypes generated randomly and train them to align with the original global prototypes. This design enhances the robustness of our method. The next section contains the details.

4.2 GAN Training

To enhance local training, our goal is to generate homogeneous datasets that can generalize local models. The main challenge lies in optimizing a generative model to produce meaningful virtual datasets. To achieve the goal, we employ trainable prototypes and the global model during the training of the generative model to align the features of virtual images with those of real images. The illustration of training procedure is shown in Fig.1.

Server embarks on training GAN after obtaining global prototypes and global model. In particular, before the training starts, server designs the architecture of the generative model to match the size training images. This ensures that the output of the GAN, after a view operation, can be used as the input of system model in FL. And the complexity of the GAN should be designed to match the complexity of the training dataset. We use the notation G_w to denote the generative model used in our system, which is parameterized by w. The function of model G_w is to output a virtual image for each given label, i.e. $G_w : y \to \mathcal{X}$.

Besides, it randomly initializes a set of prototypes $\mathcal{P}'$, and ensures $|P'| = C$. The virtual prototype P'_c is trainable and is constrained to approximate the real prototype P_c during training. This approach helps compensate for the absence of prototypes of certain classes. It avoids imposing strict constraints on complete classes, thereby enhancing generalization in scenarios with heterogeneous data. The optimization function for $\mathcal{P}'$ is as follows:

$$\min_{\mathcal{P}'} \mathrm{d}(\mathcal{P}', \mathcal{P}) = \sum_{c=1}^{C} \mathrm{d}(P'_c, P_c) \tag{6}$$

where,

$$\mathrm{d}(P'_c, P_c) = \begin{cases} -\log \frac{\exp(-\|P'_c - P_c\|/\tau)}{\sum_{i\in[C]} \exp(-\|P'_c - P_i\|/\tau)}, & \text{if } P_c \in \mathcal{P} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Where τ denotes distillation temperature and the function $\mathrm{d}(\cdot)$ denotes the distance between the two inputs. The virtual feature set $\mathcal{P}'$ is co-optimized alongside the generative model G_w.

The server constructs input vectors for G_w by concatenating the one-hot encoded label with random noise. The output vectors for G_w are then reviewed to generate the virtual dataset D'. Next, D' is fed into the frozen global model to obtain feature representations, which are expected to be consistent with the global prototypes. Motivated by prototype learning[25], we define a distance-based cross entropy(DCE) loss term for supervised learning, where the computation for an input $(x, y) \in D'$ is performed as follows:

$$\ell_{dce} = -\log \frac{\exp\left(-\mathrm{d}(f_\phi(x), P'_y)/\tau\right)}{\sum_{c\in[C], c\neq y} \exp\left(-\mathrm{d}(f_\phi(x_i), P'_c)/\tau\right)} \tag{8}$$

Where τ is a parameter of the distillation temperature. The objective of ℓ_{dce} is to minimize the distance between the feature vectors of x and the prototypes corresponding to label y, effectively clustering features of the same class together.

We train G_w to generate virtual images whose features closely align with the ground-truth prototypes by minimizing the loss function ℓ_{dce}. Over-fitting, however, might result from solely minimizing ℓ_{dce}. Given this, inspired by work [43], we add another loss term called prototype loss, as a regularization to enhance the generalization performance of the generative model G_w. For an input $(x, y) \in D'$, the prototype loss is defined as:

$$\ell_p = ||f_\phi(x) - P'_y||_2^2 \tag{9}$$

The prototype loss ℓ_p is integrated with the classification loss ℓ_{dce} to train the generative model. Therefore, the optimization objective for G_w is to minimize:

$$\begin{aligned}\min_{G_w,\mathcal{P}'} \mathcal{L}_G = \min_{G_w,\mathcal{P}'} E_{(x,y)\sim D'}[&\ell_{dce}(G_w,\mathcal{P}',f_\phi;(x,y)) \\ &+\lambda_s \ell_p(G_w,\mathcal{P}',f_\phi;(x,y))] + \mathrm{d}(\mathcal{P}',\mathcal{P})\end{aligned} \tag{10}$$

where the hyperparameter λ_s controls the weight of prototype loss. The co-optimization function is designed to identify optimal G_w and $\mathcal{P}'$ for generating virtual datasets that align with the system-wide feature distributions.

The design of $\mathcal{L}_G$ offers three key advantages: (1) The introduction of a trainable virtual prototype set not only addresses the issue of missing classes, but also allows virtual vectors to adjust to more optimal positions during co-optimization with the generator; (2) The distance-based loss term ℓ_{dce} optimizes the generator to produce virtual images, whose embeddings are mapped into the same metric space as the essential semantics of multiple local data $x's$. In other words, the distribution of the generated virtual dataset approximates the consensus distribution of the training dataset from a global perspective; (3) The regularization term ℓ_p helps draw features closer to their corresponding prototypes. It effectively enables the generator to produce virtual data with more compact intra-class features and greater inter-class separability;

Once the server has trained the generator, it is used to generate independent and equally distributed virtual datasets $\mathcal{D}'$. And at the start of the next round, this dataset is distributed to the selected clients along with the global model. Next, we will explain how clients leverage virtual datasets to support their local training.

4.3 Local Objective

Client initiates local training as soon as they receive the global model and virtual dataset. In our system, alongside training models with local datasets, clients also utilize virtual datasets for feature alignment and classifier calibration. It helps mitigate the client-drift issue and results in more generalized local models. The local loss is illustrated in Fig.4, where both the local dataset and the virtual dataset are used for supervised training in the local model, with the distance between their features also being constrained.

Specifically, to align features, we constrain the distance between the representations of real images and those of virtual images with the same labels. As stated in Section 3.2, in non-IID data scenarios, different clients' feature extractors may map images of the same class to distinct feature vectors, which indicates that the performance of local models is negatively impacted[44]. Therefore, we employ virtual datasets with class-clustered features to align the feature domain among local models. What's more, it bridges the gap between clients' local data by providing a unified representation of the features from a global data perspective. For client k, the feature alignment loss ℓ_{fa} for an input $(x,y) \in D_k$ is defined by

$$\ell_{fa} = -\log \sum_{(x',y)\in D'} \frac{\exp(f_\phi(x)\cdot f_\phi(x')/\tau)}{\sum_{(x',\neg y)\in D'} \exp(f_\phi(x)\cdot f_\phi(x')/\tau)} \tag{11}$$

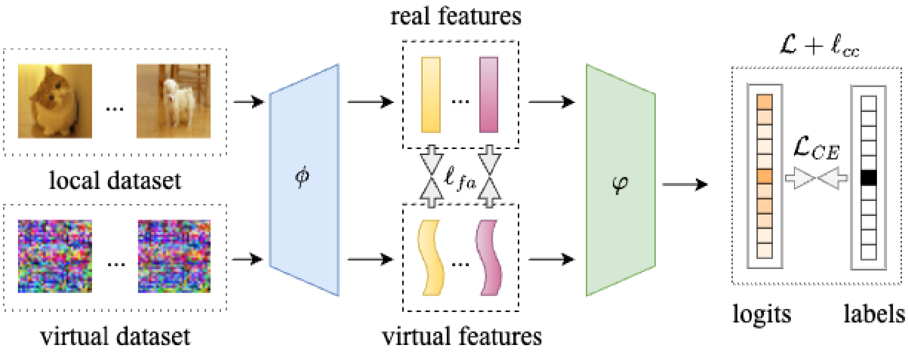


Fig. 4: The illustration of local loss in FedPG. The virtual dataset supports local training by enabling feature alignment and classifier de-biasing simultaneously.

Where the $\cdot$ sign denotes the inner product. $(x', y) \in D'$ represents the set of virtual images with label y, while $(x', \neg y) \in D'$ denotes the complement set of $(x', y) \in D'$. And τ is temperature hyperparameter. Here, we employ supervised contrastive loss[45] to minimize the distance between features with the same label while maximizing it for those with different labels.

We also introduce the virtual dataset into local training to calibrate the great bias in local classifiers. Relevant study[7] has demonstrated that even a small amount of data sharing can significantly enhance the performance of heterogeneous FL. In our framework, the homogeneous virtual datasets serve as public datasets without introducing privacy concerns. When incorporate virtual datasets into local training, the model is exposed to more isomorphic training data, leading to more balanced predictions. The classifier tuning loss ℓ_{ct} is defined by

$$\ell_{ct} = E_{(x',y')\sim D'}[\mathcal{L}(M_k; x', y')] \tag{12}$$

Where we denote the classification loss function as $\mathcal{L}$, and we use cross entropy loss in our experiments.

The objective of client k's local training is to minimize

$$\begin{aligned} \min_{M_k} \mathcal{L}_k &= E_{(x,y)\sim D_k}[\mathcal{L}(M_k; x, y)] + E_{(x',y')\sim D'}[\mathcal{L}(M_k; x', y')] \\ &+ \lambda_c E_{(x,y)\sim D_k,(x',y')\sim D'} \ell_{fa}(M_k, x|y, x'|y) \end{aligned} \tag{13}$$

Where the hyperparameter λ_c controls the weight of feature alignment loss.

In our algorithm, we introduce both feature alignment loss and classifier calibration loss into the local objective, offering three key benefits: (1) We align feature domains by leveraging the virtual dataset to help learn more generalized local models. Furthermore, in cases where certain local classes are absent, the virtual data can mitigate the shift in the corresponding feature domains; (2) The homogeneous virtual dataset introduces consistency across data from different clients, helping to align the

Algorithm 1: the FedPG framework

```
Input: global model $M^0 = \{\phi, \varphi\}$, global dataset $D$, virtual dataset $D'$, prototypes set $\mathcal{P}$,
       generator $G_w$.
Server executes:
initialize $M^0$, $G_w$
for each round $r = 0, 1, ..., R-1$ do
    for each client $k \in [K]$ do
        send $M^r$ and $\mathcal{D}'$ to client $k$
        $M_k^r$, $\mathcal{P}_k \leftarrow$ LocalTraining($M^r, \mathcal{D}', k$)
    end
    // aggregation operations
    $M^{r+1} \leftarrow \sum_{k=1}^{K} \frac{|D_k|}{|D|} M_k^r$
    $\mathcal{P} \leftarrow \mathrm{avg}(\mathcal{P}_1, ..., \mathcal{P}_K)$, i.e. Eq.5
    // virtual dataset updating
    $D' \leftarrow$ GANTraining($G_w, \mathcal{P}, \phi^{r+1}$)
end
Output: the final global model $M^R$.
```

output of local classifiers with the distribution of global data. (3) From a model-wide perspective, our strategy enhances the similarity among local models and mitigates the client drift issue. It ultimately benefits the performance of aggregated model, i.e. the global model. This creates a positive feedback loop between the generalized local and global models.

4.4 Overall framework of FedPG

Algorithm 1 outlines the complete procedure of our FedPG algorithm. For simplicity, we present our framework without incorporating a specific client selection technique in Algorithm 1. Due to the use of virtual prototypes, there are no additional constraints on the client selection strategy.

In each round, the server distributes the global model and virtual dataset to the selected clients, who then return their local models and prototypes set. The server performs separate aggregation operations on the models and prototypes. Finally, the server trains the generator to update the virtual dataset for the subsequent round. For the **LocalTraining** function, the objective is defined in Eq. 13. Each client updates the global model using both local and virtual datasets through stochastic gradient descent during local training. At the end of the process, clients compute and return local prototypes along with the updated model. The **GANTraining** function, detailed in Section 4.2, is responsible for training the generator, thereby updating the virtual dataset. This mechanism ensures that the virtual dataset remains aligned with the global model throughout the training process.

Our framework first incorporates a class-clustered virtual dataset to simultaneously achieve feature alignment and classifier de-biasing, enhancing robustness in class-missing data scenarios. Additionally, it naturally preserves client privacy, as reconstructing raw images from the shared prototypes is inherently infeasible. However, the incorporation of virtual datasets comes with additional resource costs, which we analyze through experiments in the following.

5 Experimental Evaluation

5.1 Experiment Setup

Dataset and model. We conducted evaluations on three datasets: CIFAR-10, CIFAR-100[31], and FEMNIST[32]. Furthermore, to assess algorithms' performance under different non-IID scenarios, we simulate diverse non-IID data distributions using three partitioning techniques. These techniques are employed to allocate the dataset among clients as their local datasets: (1) Latent Dirichlet Sampling(LDA)[46]. LDA is a widely used method for partitioning data in FL. It employs the Dirichlet distribution with parameter α to assign a portion of samples from each class to different clients. We implement LDA with $\alpha = 0.05, 0.1$. (2) 2-class Division. Following the approach in [19, 33], each client is assigned data from two distinct classes. Specifically, we first split each class into two equal parts and then randomly assign two parts to each client. This represents a typical class-missing data scenario. (3) Subset Partition. Similar to prior works[7, 19], we ensure that all clients access to samples from every class, but with a dominant class. Each client has a primary class comprising 95% of its local data, while the remaining classes are distributed evenly. Additionally, for CIFAR-100, we apply LDA with $\alpha = 0.1$. Since FEMNIST is inherently a federated dataset organized by user, no further partitioning is needed.

For global model, We employ ResNet-18 and ResNet-50[47] as feature extractors for CIFAR-10 and CIFAR-100, respectively, with a single-layer fully connected classifier. For FEMNIST, we adopt a CNN network consisting of five 5x5 convolution layers as the feature extractor, followed by two fully connected layers with ReLU activation as the classifier. Besides, we use a simple five-layers fully connected network with LeakyReLU activation as the default generator.

System Setting. We conduct our experiments using the FederatedScope platform[48]. In each round, we randomly select clients to participate in training with a proportion of s. For CIFAR-10, we configurate the total number of joined clients is 10, and sample half of them, i.e. $s = 0.5$. For CIFAR-100, we utilize 100 clients with a sampling ratio of $s = 0.1$. And we employ 200 clients with a sampling ratio of 0.05 for FEMNIST. For each classification task, the total number of communication rounds and local update epochs are set as follows: CIFAR-10: 100 rounds, 5 epochs, CIFAR-100: 200 rounds, 3 epochs, FEMNIST: 400 rounds, 1 epochs. The batch size for all datasets is set to 64. For a fair comparison, we use identical StepLR scheduler and SGD optimizer parameters across all algorithms. The generator is trained for 300 epochs at the server with a temperature of $\tau = 2$, a prototype loss weight of $\lambda_s = 0.001$, and a learning rate of 2×10^{-5}. We set the virtual dataset to contain 64 samples per class. For local training, the feature alignment weight is set to $\lambda_c = 1$.

Baselines. We evaluate our approach, FedPG, against FedAvg and several other methods designed to address non-IID issues in FL, including the two categories outlined in Section 2: (1) Alleviating data heterogeneity: VHL[19]. (2) Enhancing model consistency: FedProx[8] and FedNTD[35] (whole model), MOON[37] (feature extractor), FedGen[17] and CCVR[16] (classifier). We evaluate each method using three different parameter settings or parameter combinations and report the best-performing configuration. Additionally, since CCVR[16] is applied post-training, we evaluate it using the optimal global model obtained from other baseline methods.

5.2 Experiment Result

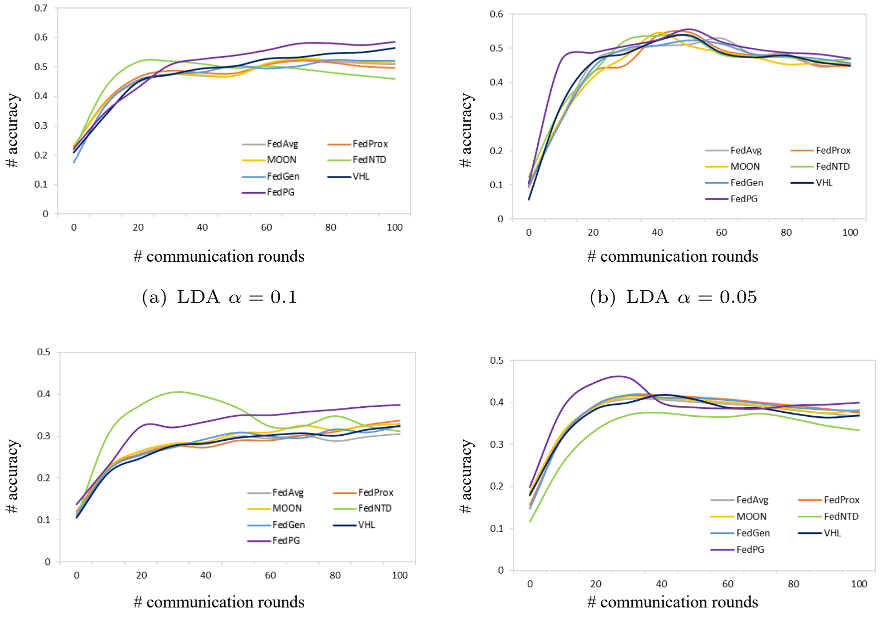
Main Results. We evaluate the performance of FedPG and baseline methods using two metrics: final accuracy and the ratio of communication rounds required to reach a target accuracy compared to FedAvg. The main results are summarized in Table 1. The table shows that our FedPG outperforms all baselines in both model performance and convergence speed, reaching state-of-the-art results in most scenarios. The Fig.5 presents the average accuracy at each communication round on CIFAR-10 under different partition strategies. It is evident that our method achieves the fastest convergence and requires the least amount of communication rounds to achieve the same classification accuracy. Specifically, compared to FedAvg, improves accuracy by an average of 3.76% while reducing the number of communication rounds by half. Although our approach introduces additional computational and downlink communication overhead, it compensates for these costs with significantly accelerated convergence. We will elaborate on this in the following section.

Impacts of non-IID. Table 1 shows that FedPG is less affected by non-IID data compared to other methods, maintaining superior performance across varying degrees of data heterogeneity. Additionally, the results indicate that the other two types of non-IID partitioning, 2-class and subset, significantly degrade the generalization performance of FL. Even in these more challenging settings, FedPG consistently outperforms other methods, demonstrating its robustness in handling various forms of data heterogeneity.

Visualization. To evaluate the effectiveness of our approach in enhancing model generalizability, we employ t-SNE [42] to visualize the feature distributions of global models trained on CIFAR-10 ($\alpha = 0.1$) using FedAvg and FedPG. As shown in Fig. 6 (a), FedAvg causes features from different categories to overlap, making classification more challenging and reducing the generalizability of the global model. In contrast, Fig. 6 (c) illustrates that FedPG effectively clusters features by category, demonstrating

Table 1: Comparison of Accuracy and Convergence Speed Across CIFAR-10, CIFAR-100, and FEMNIST for Different FL Methods.

Dataset	Partition	FedAvg	VHL	FedProx	FedNTD	MOON	FedGen	CCVR	FedPG
FEMNIST	none	85.99	83.79*(Nan)*	85.82*(0.9)*	84.99*(0.6)*	**86.23***(1.1)*	84.11*(0.5)*	85.92	86.16***(1.3)***
CIFAR-10	$\alpha = 0.05$	51.07	54.82*(0.9)*	52.49*(1.1)*	53.43*(0.7)*	50.71*(0.6)*	52.35*(1.2)*	54.20	**56.10*****(1.4)***
	$\alpha = 0.1$	56.08	57.32*(1.1)*	55.67*(2.4)*	57.68***(3.9)***	56.42*(3.2)*	57.16*(1.6)*	57.31	**59.67***(1.5)*
	2-class	37.44	39.60*(1.0)*	37.03*(1.4)*	42.19***(2.6)***	41.37*(1.5)*	38.05*(1.4)*	**42.53**	42.37*(2.3)*
	subset	44.10	41.74*(Nan)*	44.72*(0.8)*	41.88*(Nan)*	43.69*(0.6)*	44.22***(0.9)***	43.25	**47.76***(0.6)*
CIFAR-100	$\alpha = 0.1$	31.63	30.05*(0.8)*	31.78*(1.3)*	34.38*(2.3)*	31.9*(1.6)*	30.57*(0.8)*	35.43	**36.81*****(3.8)***



(a) LDA $\alpha = 0.1$

(b) LDA $\alpha = 0.05$

(c) 2-Classes

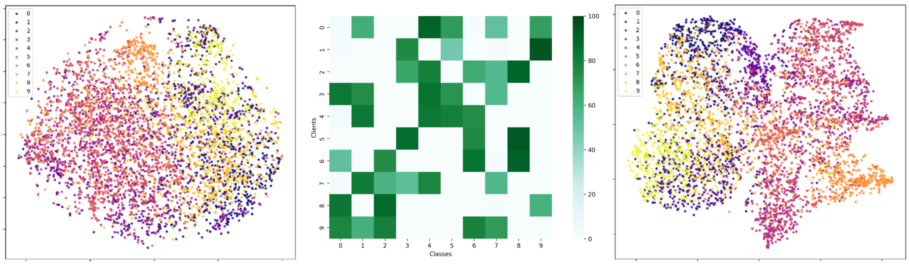
(d) Subset

Fig. 5: The average accuracy in different number of communication rounds on CIFAR-10 dataset with three categories of partition methods.

its capability to learn generalized representations and mitigate domain drift. We also provide a heatmap to illustrate the class-wise accuracy of all local models trained with FedAvg, as shown in 6 (b). The results indicate a severe imbalance in class-wise accuracy across local models. At the same time, by incorporating the virtual dataset into local training, our method calibrates the bias in the classifiers and achieves more balanced predictions.

5.3 Ablation Study

Strategy. To further validate the effectiveness of our method, we evaluate various ablation strategies, which can be categorized into two main areas: GAN training(GT) and local training(LT). The strategies are as follows: (1) GT-d: Using virtual prototypes to train the GAN without constraining the distance between virtual prototypes and real ones, i.e. removing term $\mathrm{d}(\mathcal{P}', \mathcal{P})$ from the Eq.10; (2) GT-p: Training GAN without regularization term, i.e. removing prototype loss term ℓ_p from the Eq.10; (3) LT-fa: Removing the feature alignment loss term ell_{fa} from the local training loss in Eq.13, meaning clients perform supervised learning using the union of the local dataset and virtual dataset; (4) LT-cc: Removing the classifier tuning loss term ell_{ct} from the local training loss in Eq.13, where the virtual dataset is only used to provide feature representations for alignment.



(a) t-SNE of FedAvg (b) Class-wise Accuracy (c) t-SNE of FedPG

Fig. 6: Subfigures (a) and (c) present the t-SNE visualizations of feature distributions for global models trained with FedAvg and FedPG, respectively. Subfigure (c) further illustrates the class-wise accuracy of the global model trained with FedAvg.

Baseline	Strategy	virtual images per class				
		64	128	256	512	1024
CIFAR-10 (51.07)	GT - d	54.82	56.13	54.43	55.57	54.30
	GT - p	53.16	55.60	56.17	55.82	55.24
	LT - fa	54.84	55.72	54.26	54.08	53.42
	LT - ct	54.07	54.25	53.94	53.72	54.15
	FedPG	55.39	56.55	57.04	56.20	55.43

Table 2: Ablation Study on CIFAR-10: Performance of strategies Under Different Virtual Dataset Sizes

Table 2 presents the performance of these strategies across different virtual dataset sizes. It is clear that all these strategies result in improvements over FedAvg, likely due to the incorporation of the public dataset. We can also observe that, compared to the strategies in GT, the strategies in LT have a greater impact on performance. Notably, our method consistently outperforms its variants across various sizes of the virtual dataset, demonstrating the effectiveness and robustness of our approach in enhancing FL performance under non-IID conditions.

5.4 Other Results

Parameter Configuration. To determine the optimal hyperparameter λ_c, we evaluate the global model's performance under different feature alignment weights. In the experiments, we fix the number of virtual images per class at 256 and set the training batch size for the virtual dataset to 64. Figure 7 illustrates the impact of λ_c on FedPG. Our method consistently outperforms FedAvg across a range of λ_c values, demonstrating its effectiveness in improving FL performance under non-IID conditions. Notably, the global model achieves the highest accuracy when $\lambda_c = 10$, while setting it too small or too large reduces effectiveness. This could be attributed to overly loose or overly rigid feature alignment. In summary, while FedPG is relatively robust to variations in weight λ_c, it is crucial to choose a balanced value to maximize performance.

Resource Cost. FedPG introduces GAN training and synthetic datasets, which incur additional resource costs. We measure the computational overhead of GAN training (GT) and local training (LT) in terms of energy consumption (Wh). Additionally, we record the duration of each communication round. We evaluate two different GAN architectures: fully connected layers and convolutional networks. As shown in Table 3, FedPG incurs an additional 800 Wh of computational cost on the server and 100 Wh on the clients. The overall resource consumption remains within an acceptable range. Besides, when GAN training and local training run sequentially, it adds a delay of 10-20 seconds to the communication round duration. However, this delay can be eliminated by running them in parallel.

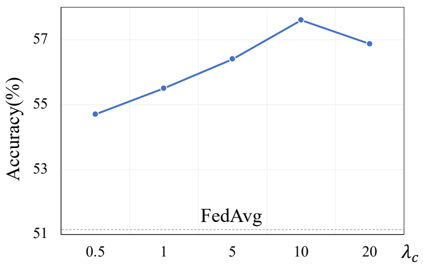


Fig. 7: The top-1 test accuracy for FedPG with different λ_c.

Method	GT	LT	Dur.(s)
FedAvg	-	269.63	2.11
FedPG+fc	822.88	367.58	15.11
FedPG+cnn	868.95	367.58	28.54

Table 3: Energy consumption and communication round duration.

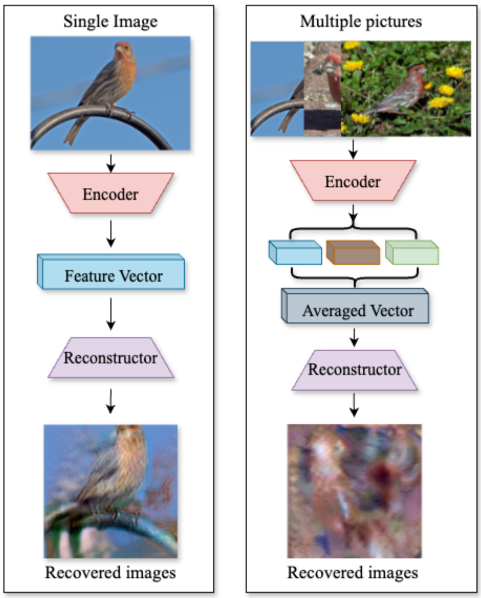


Fig. 8: Reconstruction of the original image from a feature or a prototype.

Privacy Analysis. In our framework, clients share their local prototypes to assist with GAN training on the server. A prototype is the average vector of feature representations for a given class[49]. To evaluate the privacy-friendliness of our method, we attempt to reconstruct the raw image from the prototype using a SqueezeNet model[50]. As shown in Fig. 8, the reconstructor can nearly reconstruct the raw image from a single feature vector. However, when averaged over three vectors, the reconstructor generates only a blurry image from this prototype. This demonstrates that sharing prototypes does not pose a privacy risk for clients.

6 Conclusion

In this paper, we address several key limitations of existing approaches in FL with non-IID data: their ineffectiveness in handling class missingness, the potential threats to privacy, and the partial solutions they offer to the client drift issue. To overcome these challenges, we propose a universal and privacy-preserving method, FedPG, which includes a novel image generation approach and a local training strategy. Our method achieves feature alignment and classifier de-biasing, enhancing the generalization capability of local models, which ultimately benefits the global models. Our comprehensive experimental results demonstrate that FedPG significantly boosts performance under a variety of non-IID settings. We hope that further research will delve into data and model compression to alleviate the additional overheads, thereby facilitating the wider adoption of our method.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 62206238, 62176122, 62061146002), the Natural Science Foundation of Jiangsu Province (No. BK20220562), the Natural Science Research Project of Universities in Jiangsu Province (No. 22KJB520010), the China Postdoctoral Science Foundation (No. 2023M732985).

Data availability statement

The data and models in this study can be requested from the corresponding author. The code is publicly accessible at https://github.com/XueBaolu/FedPG/.

Conflict of interest

The authors have no relevant financial or non-financial interests to disclose.

References

[1] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., *et al.*: Imagenet large scale visual recognition challenge. International journal of computer vision **115**, 211–252 (2015)

[2] Xiong, H., Yan, H., Obaidat, M.S., Chen, J., Cao, M., Kumar, S., Agarwal, K., Kumari, S.: Efficient and privacy-enhanced asynchronous federated learning for multimedia data in edge-based iot. ACM Transactions on Multimedia Computing, Communications and Applications (2024)

[3] Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)

[4] Konečnỳ, J., McMahan, H.B., Ramage, D., Richtárik, P.: Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527 (2016)

[5] Shenaj, D., Rizzoli, G., Zanuttigh, P.: Federated learning in computer vision. IEEE Access (2023)

[6] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., *et al.*: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021)

[7] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. arXiv preprint arXiv:1806.00582 (2018)

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

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

[10] Dai, Y., Chen, Z., Li, J., Heinecke, S., Sun, L., Xu, R.: Tackling data heterogeneity in federated learning with class prototypes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 7314–7322 (2023)

[11] Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International Conference on Machine Learning, pp. 5132–5143 (2020). PMLR

[12] Huang, W., Ye, M., Du, B.: Learn from others and be yourself in heterogeneous federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10143–10153 (2022)

[13] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607 (2020). PMLR

[14] Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15750–15758 (2021)

[15] Moreno-Torres, J.G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern recognition **45**(1), 521–530 (2012)

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

[17] Zhu, Z., Hong, J., Zhou, J.: Data-free knowledge distillation for heterogeneous federated learning. In: International Conference on Machine Learning, pp. 12878–12889 (2021). PMLR

[18] Hao, W., El-Khamy, M., Lee, J., Zhang, J., Liang, K.J., Chen, C., Duke, L.C.: Towards fair federated learning with zero-shot data augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3310–3319 (2021)

[19] Tang, Z., Zhang, Y., Shi, S., He, X., Han, B., Chu, X.: Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In: International Conference on Machine Learning, pp. 21111–21132 (2022). PMLR

[20] Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.-L.: Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. arXiv preprint arXiv:1811.11479 (2018)

[21] Luo, B., Xiao, W., Wang, S., Huang, J., Tassiulas, L.: Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In: IEEE INFOCOM 2022-IEEE Conference on Computer Communications, pp. 1739–1748 (2022). IEEE

[22] Li, C., Zeng, X., Zhang, M., Cao, Z.: Pyramidfl: A fine-grained client selection framework for efficient federated learning. In: Proceedings of the 28th Annual International Conference on Mobile Computing And Networking, pp. 158–171 (2022)

[23] Ghosh, A., Chung, J., Yin, D., Ramchandran, K.: An efficient framework for clustered federated learning. Advances in Neural Information Processing Systems **33**, 19586–19597 (2020)

[24] Zeng, S., Li, Z., Yu, H., He, Y., Xu, Z., Niyato, D., Yu, H.: Heterogeneous federated learning via grouped sequential-to-parallel training. In: International Conference on Database Systems for Advanced Applications, pp. 455–471 (2022). Springer

[25] Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)

[26] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)

[27] Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: Fedproto: Federated prototype learning across heterogeneous clients. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 8432–8440 (2022)

[28] Wang, L., Zhang, Q., Sang, L., Wu, Q., Xu, M.: Federated prototype-based contrastive learning for privacy-preserving cross-domain recommendation. arXiv preprint arXiv:2409.03294 (2024)

[29] Huang, W., Ye, M., Shi, Z., Li, H., Du, B.: Rethinking federated learning with domain shift: A prototype view. in 2023 ieee. In: CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16312–16322 (2023)

[30] Jiang, N., Fang, J., Xu, J., Shao, Y.: Ssd based on contour–material level for domain adaptation. Pattern Analysis and Applications **24**(3), 1221–1229 (2021)

[31] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

[32] Caldas, S., Duddu, S.M.K., Wu, P., Li, T., Konečnỳ, J., McMahan, H.B., Smith, V., Talwalkar, A.: Leaf: A benchmark for federated settings. arXiv preprint arXiv:1812.01097 (2018)

[33] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282 (2017). PMLR

[34] Song, Y., Liu, H., Zhao, S., Jin, H., Yu, J., Liu, Y., Zhai, R., Wang, L.: Fedadkd: heterogeneous federated learning via adaptive knowledge distillation. Pattern Analysis and Applications **27**(4), 134 (2024)

[35] Lee, G., Jeong, M., Shin, Y., Bae, S., Yun, S.-Y.: Preservation of the global knowledge by not-true distillation in federated learning. Advances in Neural Information Processing Systems **35**, 38461–38474 (2022)

[36] Yu, F., Zhang, W., Qin, Z., Xu, Z., Wang, D., Liu, C., Tian, Z., Chen, X.: Fed2: Feature-aligned federated learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 2066–2074 (2021)

[37] Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10713–10722 (2021)

[38] Li, Z., Shang, X., He, R., Lin, T., Wu, C.: No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5319–5329 (2023)

[39] Chang, H., Shejwalkar, V., Shokri, R., Houmansadr, A.: Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. arXiv preprint arXiv:1912.11279 (2019)

[40] Seo, E., Niyato, D., Elmroth, E.: Resource-efficient federated learning with non-iid data: An auction theoretic approach. IEEE Internet of Things Journal **9**(24), 25506–25524 (2022)

[41] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8110–8119 (2020)

[42] Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)

[43] Yang, H.-M., Zhang, X.-Y., Yin, F., Liu, C.-L.: Robust classification with convolutional prototype learning. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3474–3482 (2018). https://doi.org/10.1109/CVPR.2018.00366

[44] Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2010) https://doi.org/10.1109/TKDE.2009.191

[45] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Advances in neural information processing systems **33**, 18661–18673 (2020)

[46] Hsu, T.-M.H., Qi, H., Brown, M.: Measuring the effects of non-identical data distribution for federated visual classification. arXiv preprint arXiv:1909.06335 (2019)

[47] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[48] Wang, Z., Kuang, W., Xie, Y., Yao, L., Li, Y., Ding, B., Zhou, J.: Federatedscope-gnn: Towards a unified, comprehensive and efficient package for federated graph learning. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4110–4120 (2022)

[49] Allen, K., Shelhamer, E., Shin, H., Tenenbaum, J.: Infinite mixture prototypes for few-shot learning. In: International Conference on Machine Learning, pp. 232–241 (2019). PMLR

[50] Iandola, F.N.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)