AI Open 6 (2025) 82–92

Contents lists available at ScienceDirect

AI Open

journal homepage: www.keaipublishing.com/en/journals/ai-open

Full length article

FedGPA: Federated Learning with Global Personalized Aggregation

Zongfu Han [a], Yu Feng [a], Yifan Zhu [a,*], Zhen Tian [a], Fangyu Hao [a], Meina Song [a,b]

[a] School of Computer Science(National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing, 100876, China
[b] China University of Petroleum - Beijing at Karamay, Karamay, 834000, China

ARTICLE INFO

Keywords:
Personalized federated learning
Data heterogeneity
Federated aggregation

ABSTRACT

A significant challenge in Federated Learning (FL) is addressing the heterogeneity of local data distribution across clients. Personalized Federated Learning (PFL), an emerging method aimed at overcoming data heterogeneity, can either integrate personalized components into the global model or train multiple models to achieve personalization. However, little research has simultaneously considered both directions. One such approach involves adopting a weighted aggregation method to generate personalized models, where the weights are determined by solving an optimization problem among different clients. In brief, previous works either neglect the use of global information during local representation learning or simply treat the personalized model as learning a set of individual weights. In this work, we initially decouple the model into a feature extractor, associated with generalization, and a classifier, linked to personalization. Subsequently, we conduct local–global alignment based on prototypes to leverage global information for learning better representations. Moreover, we fully utilize these representations to calculate the distance between clients and develop individual aggregation strategies for feature extractors and classifiers, respectively. Finally, extensive experimental results on five benchmark datasets under three different heterogeneous data scenarios demonstrate the effectiveness of our proposed FedGPA.

1. Introduction

Contemporary learning tasks predominantly rely on deep neural networks (DNNs), which necessitate substantial volumes of training data to achieve satisfactory model performance (LeCun et al., 2015). However, data are usually dispersed among different parties in practice (e.g., data silos and mobile devices). The growing concern over privacy issues and the implementation of pertinent regulations have led to increasing challenges and costs associated with the direct acquisition and centralization of data for learning and training purposes (Andreina et al., 2021; Aono et al., 2017). To address the above challenge, Federated Learning (FL), as a promising machine learning approach, enables distributed clients to collaboratively train a global model without accessing their data by sharing their local model parameters for aggregation (Kairouz et al., 2021). The Federated Averaging (FedAvg) algorithm is a popular federated learning approach (McMahan et al., 2017). In each FedAvg round, a subset of clients is chosen for local training, and their results are subsequently aggregated on the server based on sample weights. FL algorithms have found extensive applications in various real-world scenarios, including medical imaging (Adnan et al., 2022), object detection (Liu et al., 2020), and predicting users' next-word (Hard et al., 2018).

However, the conventional FL approach encounters several fundamental challenges: (a) poor performance and convergence when dealing with highly heterogeneous data, and (b) absence of personalized solutions (Asad et al., 2020; Bietti et al., 2022). For instance, the next-word prediction model trained by FedAvg for users may not always be effective due to their personalized habits. A singular global model of this nature can deviate significantly from individual optimal models. Besides, it has been reported in Karimireddy et al. (2020) that a drift in the updates of each client caused by data heterogeneity resulting in poor performance and unstable convergence. These challenges deteriorate the performance of the global FL model on individual clients in the presence of heterogeneous local data distributions, and may even discourage affected clients from participating in the FL process (Tan et al., 2022). To tackle these problems, Personalized Federated Learning (PFL) has been proposed and researched by the FL community (Dai et al., 2022; Guo et al., 2022; Feng et al., 2025). There are two main approaches: (a) adjusting the local model on the client side and (b) refining the global model on the server side. Under the first approach, SCAFFOLD (Karimireddy et al., 2020) leverages control variates to address client drift; and MOON (Li et al., 2021) aligns the intermediate

* Corresponding author.
E-mail addresses: michan325@bupt.edu.cn (Z. Han), fydannis@bupt.edu.cn (Y. Feng), yifan_zhu@bupt.edu.cn (Y. Zhu), tianzhentz@bupt.edu.cn (Z. Tian), haofangyu@bupt.edu.cn (F. Hao), mnsong@bupt.edu.cn (M. Song).

https://doi.org/10.1016/j.aiopen.2025.03.001
Received 26 October 2024; Received in revised form 13 January 2025; Accepted 25 March 2025
Available online 15 April 2025

outputs of local and global models to reduce variance among optimized local models. In contrast, along the second approach, FedDF (Lin et al., 2020) and FedFTG (Zhang et al., 2022) introduce an additional fine-tuning step to further refine the global model; and FedPAC (Xu et al., 2023) combines a shared global feature extractor with a customized classifier during the training process.

Despite all these diverse efforts, few works consider both adjusting local model at the client side and optimizing the aggregation weight at the server side. In this paper, we propose a novel method called FedGPA, which integrates two key components: a prototype-based regularization loss to adjust the local models and a personalized aggregation strategy that calculates client-specific aggregation weights on the server side to achieve optimal personalization. We specifically tackle two pivotal challenges related to data heterogeneity: heterogeneity in category distribution and heterogeneity in sample numbers, encompassing variations in both category distribution and the total number of samples among clients. For instance, some clients may have a surplus of samples in category A but a scarcity in category B, while others may exhibit the opposite trend. Simultaneously, there exist notable differences in the overall number of samples among these clients. This scenario is frequently encountered in practice, as clients independently collect local data and often exhibit biased labelling preferences.

In our framework, we follow the setup from existing works (Oh et al., 2021; Xu et al., 2023) of dividing the model into feature extractors and classifiers. Different from these works, we design different aggregation strategies for feature extractors and classifiers to better learn global information and personalized information.

In summary, our main contributions in this paper can be summarized as follows:

- We propose a new personalized federated learning framework called FedGPA. On the client side, it adjusts the local model by incorporating a prototype-based regularization loss during local training, which helps integrate global information into local models more effectively.
- On the server side, it divides the local model into two components: feature extractors and classifiers. Different aggregation strategies are applied to these modules, enabling the calculation of personalized aggregation weights for each client to enhance model performance and personalization.
- We provide a detailed explanation of why different aggregation strategies are applied to feature extractors and classifiers, along with the specific rationale behind these strategies.
- By conducting experiments using five real datasets, we demonstrate that FedGPA outperforms other methods under diverse scenarios.

2. Related work

2.1. Federated learning under NON-IID settings

Many previous studies aim to enhance the performance of FL under non-IID data settings. These studies can be broadly categorized into two main types of methods: (a) methods that focus on adjusting the local model on the client side, referred to as single-model PFL methods, and (b) methods that focus on refining the global model on the server side, referred to as multi-model PFL methods.

Most single-model PFL methods are extensions of conventional FL algorithms (e.g., FedAvg (McMahan et al., 2017)). These methods focus on adjusting the local model during training on the client side, aiming to minimize the differences between local models. For example, FedProx (Li et al., 2020) incorporates a proximal term to the local training loss function to keep updated parameter close to the global parameters issued in the previous round. SCAFFOLD (Karimireddy et al., 2020) introduces control variates to correct the drift in local updates. MOON (Li et al., 2021) incorporates contrastive loss into federated learning to further align the features of the local and global models. An alternative single-model approach for PFL is based on meta-learning, which leverages the ability to learn a model initialization that can be easily adapted to diverse tasks or data distributions. Recent works (Lee et al., 2023; Chen et al., 2024) extended Model Agnostic Meta-learning (MAML) for FL under non-IID data distributions. For example, pFedMe (T. Dinh et al., 2020) proposes a federated meta-learning formulation using Moreau envelopes. However, the personalized learning ability of single-model methods remains limited, as it is challenging to effectively fit all heterogeneous data distributions with a single model.

Multi-model methods outperform single-model approaches by training multiple models that better adapt to the personalized needs of clients. Clustered FL (Ghosh et al., 2020; Ruan and Joe-Wong, 2022; Fu et al., 2023) assumes that clients can be partitioned into multiple clusters, with clients grouped based on loss values or gradients. A customized model is then trained for each cluster. However, these methods require great computation capability of the server with much more computation cost (e.g. FedHC (Briggs et al., 2020) requires additional communication rounds to ensure better clustering.)

In addition to clustering clients, several other PFL methods have been developed, such as those leveraging Gaussian processes (Chen and Chao, 2021), knowledge transfer (Yao et al., 2024), and additive mixtures of local and global models (Collins et al., 2021). For example, FedAMF (Yao et al., 2024) addresses knowledge forgetting by adaptively fusing global, local, and historical knowledge. However, these approaches often require access to publicly shared data or a set of inducing points. Another way to achieve PFL is by customizing the weights for model aggregation for each client. For example, FedPAC (Xu et al., 2023) employs a similarity calculation formula to customize personalized classifiers for each client with weighted coefficients. FedDWA (Liu et al., 2023) exploits the data of clients with similar distributions to enhance the accuracy of personalized models. These customized weights reflect the potential similarities between clients, sharing a similar concept to our approach. In contrast, we decompose the model and adopt a more refined aggregation strategy for each individual module.

2.2. Prototype learning

The concept of prototypes, which represents the mean of multiple feature vectors, has proven useful in a variety of tasks. In image classification, a prototype serves as a representative of a class and is calculated as the mean of feature vectors within that class (Snell et al., 2017). Prototype-based methods have also been extended to FL in several ways. For example, (Michieli and Ozay, 2021) proposes a prototype-based attention mechanism during global aggregation, aiming to enhance the importance of certain features in the aggregation process. Similarly, FedProto Tan et al. (2022) utilizes prototypes to integrate global information into the local optimization process, allowing for more efficient model updates under non-IID settings. FedProc (Mu et al., 2023) uses prototypes from both the current and previous communication rounds to design a contrastive loss, which helps to align the local models with the global one more effectively. Inspired by these works, we design a regularization loss using both local and global prototypes to incorporate global information into the local training process. Meanwhile, we use the prototype matrix between clients as a basis for calculation, which serves as one of the determining factors for personalized aggregation (see Table 1).

3. Problem formulation

Suppose there are m clients and a central server, where all clients communicate with the server to collaboratively train personalized models under data preservation policies. Each client i owns a private dataset, where the data distribution is given by $P_i(x, y)$ with input

Table 1
Notations of symbols and their descriptions used in this paper.

Notations	Descriptions
m	The number of clients
K, k	The number of classes; the number of domain classes for each client
$x, y, P_i(x, y)$	Input features, corresponding labels, and the data distribution for client i
$\mathcal{D}_i, \lvert\mathcal{D}_i\rvert$	The local dataset of client i; the total number of samples owned by the ith client
$f(\cdot)$, $c(\cdot)$, w^f, w^c	The feature extractor, classifier, and their corresponding parameters
z	The vector obtained after embedding the input features xx through the feature extractor $f(\cdot)$
$\mathbf{W}$	The collection of model parameters from all clients
$C_i^{(k)}$, $\bar{C}^{(k)}$	The local prototype and the global prototype for class k on the ith client
λ	A hyper-parameter used to balance supervised loss and regularization loss
μ	A hyper-parameter used to balance the influence of prototype similarity and sample size
$P_{i,j}$, $\hat{P}_{i,j}$	The euclidean distance between the ith and the jth clients; the inverse of $P_{i,j}$
$\alpha_{i,j}$	The weighted factor of the jth client when the ith client aggregates the feature extractor
$\beta_{i,j}$	The weighted factor of the jth client when the ith client aggregates the classifier
p_k	The proportion of class k in the local dataset
δ_i^2	The intra-class variance of client i
$\mathbf{F}_k$	The feature matrix composed of the hidden vectors z of all samples in class k
T	The number of communication rounds
E	The number of local training epochs
n	The number of samples each client possesses
s	The skewness of the class distribution

features x and corresponding labels y, classified into K classes. The data distribution between clients i and j differs, both in terms of label distribution and the number of samples each client possesses. We define g as the sequential combination of a feature extractor f, parameterized by w_i^f, that maps input features x to latent vectors, and a classifier c, parameterized by w_i^c, that maps latent vectors to output predictions. Formally, this is written as $g := f \circ c$, with $w_i := [w_i^f; w_i^c]$. Let $\ell(w_i^f, w_i^c; x, y)$ be the loss function for client i, evaluated on a data sample (x, y) drawn from its distribution $P_i(x, y)$. The global optimization objective of PFL can be formulated as:

$$\min_{\mathbf{w}} \left\{ F(\mathbf{W}) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{(x,y)\sim P_i} \left[\ell(w_i^f, w_i^c; x, y) \right] \right\} \tag{1}$$

where $\mathbf{W} = \{w_1, w_2, \ldots, w_m\}$ represents the collection of model parameters from all clients. In practice, the true data distribution $P_i(x, y)$ is unknown, and empirical risk minimization is used. Assume client i has sampled n_i data from its distribution, denoted by $\mathcal{D}_i = \{(x_i^l, y_i^l) \mid l = 1, 2, \ldots, n_i\}$, representing the empirical distribution of P_i. The empirical training objective for personalized models can then be rewritten as:

$$w^* = \arg\min_{w} \frac{1}{m} \sum_{i=1}^{m} \left[\mathcal{L}_S \left(w_i^f, w_i^c; \mathcal{D}_i \right) + \mathcal{L}_R(w_i^f, w_i^c; \Omega) \right] \tag{2}$$

The term $\mathcal{L}_S \left(w_i^f, w_i^c; \mathcal{D}_i \right) = \frac{1}{n_i} \sum_{l=1}^{n_i} \ell(w_i^f, w_i^c; x_l^i, y_l^i)$ denotes the empirical loss computed over the local dataset $\mathcal{D}_i$. The term $\mathcal{L}_R(w_i^f, w_i^c; \Omega)$ refers to a regularization term to avoid over-fitting, with Ω representing some form of global or local constraint, such as the L2 regularization that penalizes large weights.

Traditional Federated Learning (FL) aims to find the shared optimal global model w^* by solving Eq. (2). However, under heterogeneous settings, we are more interested in computing the optimization of personalized models for individual clients. We then wish to find the optimal weights $w^* = \{w_1^*, w_2^*, \ldots, w_m^*\} = \{[w_1^{f^*}; w_1^{c^*}], [w_2^{f^*}; w_2^{c^*}], \ldots, [w_m^{f^*}; w_m^{c^*}]\}$ that optimize for the client's objective, minimizing $F(\mathbf{W})$. Here $w_i^\star$ represents the optimal model for the ith client. Under IID data settings, the optimal model of any client is very close, which implies $w_i^\star \approx w_j^\star$. However, under non-IID data settings, as investigated in Kairouz et al. (2021), a general global model cannot achieve the best performance on all clients because the optimal solutions for different clients are often inconsistent. The notations and descriptions of the symbols used in this paper are presented in Table 1.

4. Proposed framework

In the previous section, it was highlighted that training uniform w^* for all clients cannot meet personalized requirements under non-IID data settings. In this section, we elaborate on the design of FedGPA and prove its effectiveness through analysis. We decouple the model into a feature extractor and a classifier, training personalized models for individual clients by implementing customized aggregation weights. The framework of our proposed FedGPA is depicted in Fig. 1.

Algorithm 1 The FedGPA framework

Input: local datasets $\mathcal{D}_i$, number of communication rounds T, number of clients m, number of local epochs E, hyper-parameter λ and μ.
Output: The final personalized global models for each client $\{[w_1^{f^*}; w_1^{c^*}], [w_2^{f^*}; w_2^{c^*}], \ldots, [w_m^{f^*}; w_m^{c^*}]\}$.

Server Executes:
Initialize: w^0 and c^0
for $t = 0, 1, \ldots, T-1$ **do**
  for $i = 1, 2, \ldots, m$ in parallel **do**
    Send the global prototypes $\bar{C}^t$ to client i
    Send the personalized global model $[w_i^f; w_i^c]$ to client i for local training
  end for
  Collect models and local prototypes from clients
  Get the global prototypes $\bar{C}^{t+1}$ as equation (4)
  Get feature extractors $\{w_1^{f^{t+1}}, w_2^{f^{t+1}}, \ldots, w_m^{f^{t+1}}\}$ as equation (8) – (11)
  Get classifiers $\{w_1^{c^{t+1}}, w_2^{c^{t+1}}, \ldots, w_m^{c^{t+1}}\}$ as equation (12) – (16)
  Return $[w_i^{f^{t+1}}, w_i^{c^{t+1}}]$ back to $\mathcal{P}_i$, $i \in \{1, 2, \ldots, m\}$
end for
ClientLocalTraining$(i, w_i^{f^t}, w_i^{c^t}, \bar{C}^t)$:
  Update feature extractor and classifier with $[w_i^{f^t}, w_i^{c^t}]$
  Train feature extractor and classifier for E epochs
  $[w_i^{f^{t+1}}, w_i^{c^{t+1}}] \leftarrow [w_i^{f^t}, w_i^{c^t}]$
  Get local prototypes $\bar{C}^{t+1}$ as equation (3)
  Send $\{[w_i^{f^{t+1}}, w_i^{c^{t+1}}], \bar{C}^{t+1}\}$ back to server

4.1. Local–global alignment based on prototype

To facilitate the local model effectively utilizing both local and global information, we decompose the deep neural network into two distinct components: the feature extractor and the classifier. The feature extractor, implemented as a convolutional layer, focuses on learning the sample representation. Meanwhile, the classifier, represented by a fully connected layer, generates the final classification vector. The feature embedding function $f : \mathcal{X} \to \mathbb{R}^d$ is parameterized by w^f, and d is the dimension of the feature embedding. We denote $z = f(w^f; x)$

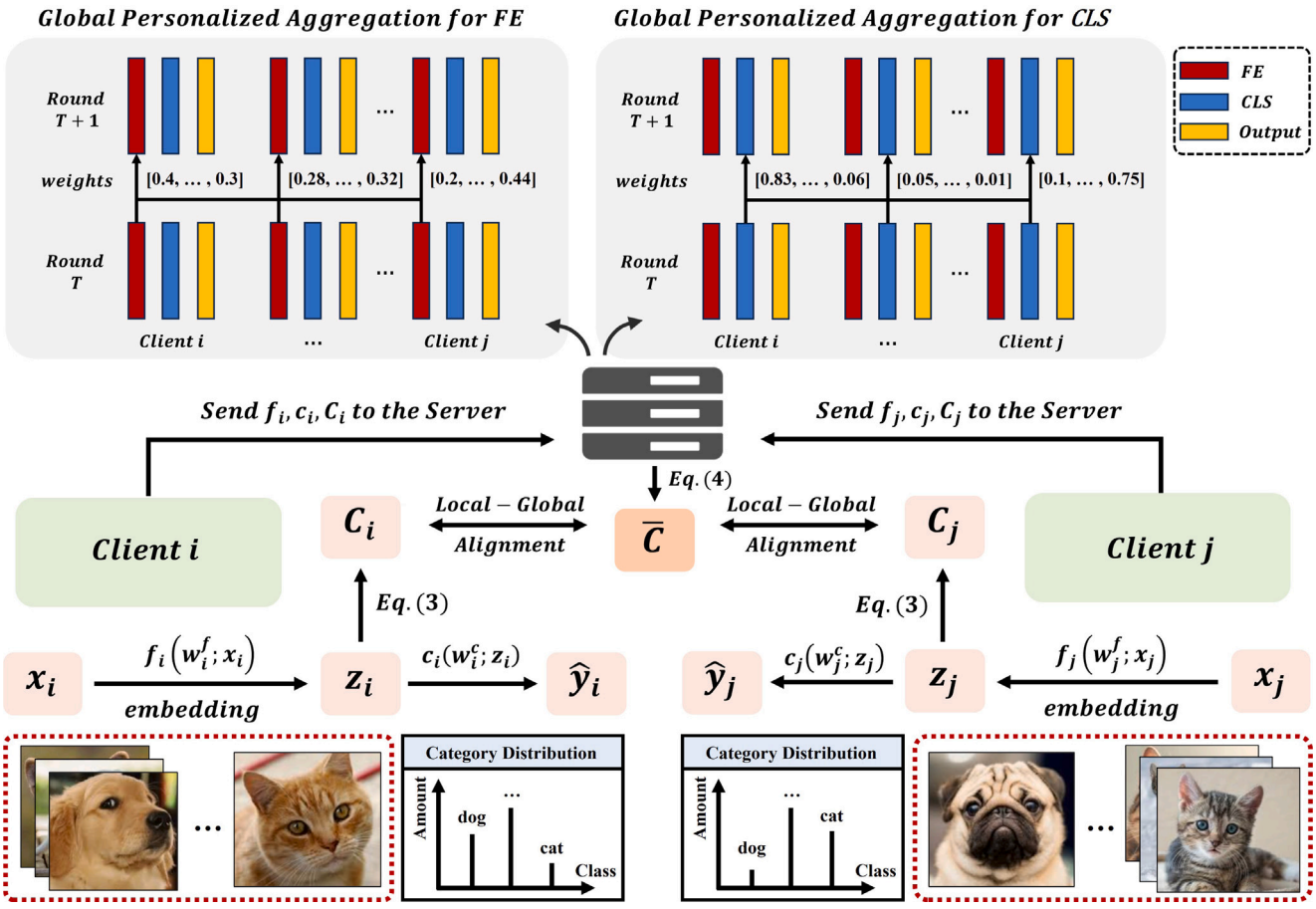

Fig. 1. The overall framework of our proposed FedGPA. $f(\cdot)$ represents the feature extractor (FE), and $c(\cdot)$ represents the classifier (CLS).

as the embeddings of x. The classification function $c : Z \to \mathbb{R}^d$ is parameterized by w^c, where d is the dimension of z. We denote $\hat{y} = c(w^c; z)$ as the final classification vector. Under the condition of data heterogeneity, the local label distribution of each client is skewed, and the data is still insufficient, which makes the local training process gradually deviate from the global optimum. A reasonable approach is to introduce a regularization term that constrains the distance between local objectives and global objectives. The hidden layer feature embedding vectors outputted by the feature extractor serve well as this target.

We introduce a regularization term into the optimization process of local training, serving as a constraint to restrict the distance between local prototype and global prototype. The local prototype is computed per class, while the global prototype is derived from the weighted average of the local prototypes from all participated clients.

4.1.1. Local prototype

For the ith client, the local prototype $C_i^{(k)}$ represents the mean value of embedding vectors of data in class k:

$$C_i^{(k)} = \frac{1}{|D_{i,k}|} \sum_{(x,y)\in D_{i,k}} f(w_i^f; x) \tag{3}$$

where $D_{i,k}$, composed of training data belonging to the kth class, is a subset of the local dataset D_i; $f(w_i^f; x)$ represents the embeddings of x.

4.1.2. Global prototype

For the kth class, the global prototype $\bar{C}^{(k)}$ represents the mean value of all local prototypes belonging to class k.

$$\bar{C}^{(k)} = \frac{1}{\sum_{i=1}^{m} |D_{i,\,k}|} \sum_{i=1}^{m} |D_{i,\,k}| \cdot C_i^{(k)} \tag{4}$$

where $C_i^{(k)}$ denotes the prototype of class k from the ith client, $|D_{i,\,k}|$ represents the number of samples of class k held by the ith client.

4.1.3. Local–global alignment

We use local prototypes and global prototypes to construct a regularization term, which allows global information to be utilized during local learning. The local regularization term is defined as follows:

$$\mathcal{R}_i(w_i^f; \bar{\mathbf{C}}) = \frac{\lambda}{|D_i|} \sum_{(x,y)\in D_i} \left\| C_i^{(k)} - \bar{C}^{(k)} \right\|_2 \tag{5}$$

where $C_i^{(k)}$ is the embeddings of the local prototype of class k, and $\bar{C}^{(k)}$ is the corresponding global prototype of class k. λ is a hyper-parameter to balance supervised loss and regularization loss.

4.2. Global personalized aggregation

Within the context of data heterogeneity, every client encounters challenges related to insufficient data and imbalanced data distribution. To enhance performance, we propose to employ personalized aggregation weights for each client in every communication round to obtain its own aggregated model, which differs from the conventional practice of utilizing a uniform global aggregation model. Moreover, unlike previous work in Xu et al. (2023), which primarily maintains a client-specific weighted average of classifiers, our emphasis lies in the personalized aggregation of both classifier and feature extractor. In the $(t+1)$-th global communication round, the aggregation function is given by:

$$w_i^{f^{t+1}} = \sum_{j=1}^{m} \alpha_{ij} w_i^{f^t}, \quad \text{s.t.} \quad \sum_{j=1}^{m} \alpha_{ij} = 1, \alpha_{i,j} \geq 0 \tag{6}$$

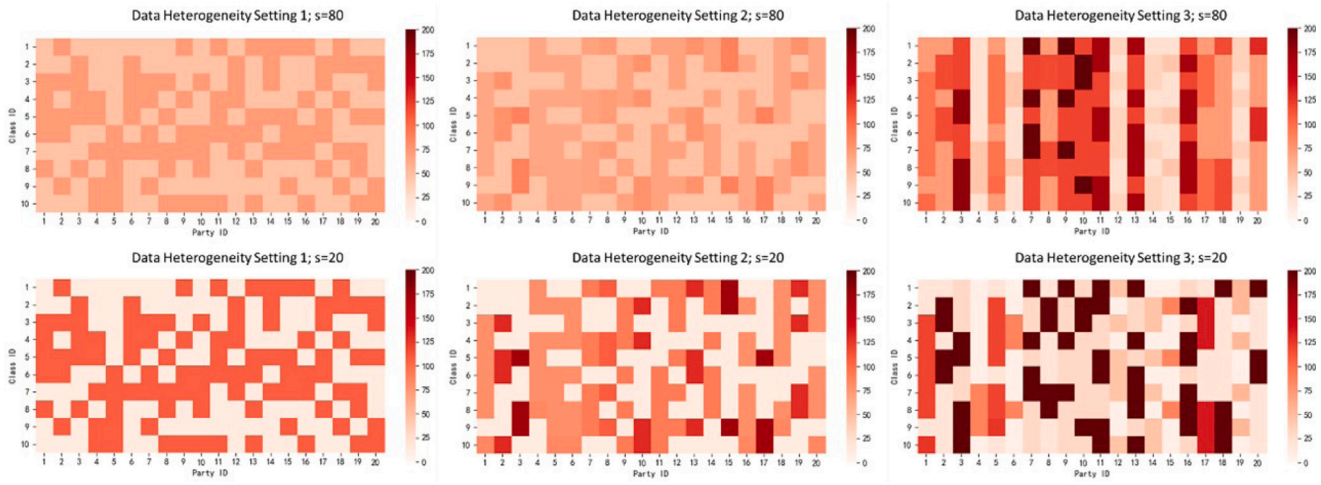

Fig. 2. The data distribution of each party under three data heterogeneity settings (taking CIFAR-10 as an example). The colour bar denotes the number of data samples. Each rectangle represents the number of data samples of a specific class in a party.

$$w_i^{c^{t+1}} = \sum_{j=1}^{m} \beta_{ij} w_i^{c^t}, \quad \text{s.t.} \quad \sum_{j=1}^{m} \beta_{ij} = 1, \beta_{i,j} \geq 0 \tag{7}$$

where α_{ij} represents the weighted factor of the jth client when the ith client aggregates the feature extractor; similarly, β_{ij} represents the weighted factor of the jth client when the ith client aggregates the classifier; with each weighted factor α_{ij}, $\beta_{ij} \geq 0$. For a better aggregation, we need to update the coefficients α and β adaptively during the training process.

4.2.1. Global personalized aggregation for feature extractor

Previous research (Oh et al., 2021; Xu et al., 2023) suggests that during global aggregation, the feature extractor should prioritize capturing global information to improve feature representation and model generalization. This approach may lead to a more uniform weight distribution, which could be advantageous. Consequently, if each client can selectively focus on aggregating information from clients with similar representations, it could continuously improve the overall model performance. This selective attention mechanism during global aggregation is crucial in ensuring that the model benefits from diverse yet relevant data, thereby contributing to its superior performance. The challenge lies in how to effectively evaluate the similarity between clients. To address this, we quantify the similarity by calculating the sum of distances between the prototypes of each class for two given clients. Let $C_i^{(k)}$ and $C_j^{(k)}$ represent the local prototypes of the ith and jth clients for category k, respectively. The distance between the ith and jth clients can then be calculated using the following formula:

$$P_{i,j} = \frac{|\mathcal{D}_{i,k}|}{|\mathcal{D}_i|} \sum_{k=1}^{K} \left\| C_i^{(k)} - C_j^{(k)} \right\|_2, \quad (x, y) \in \mathcal{D}_i \tag{8}$$

$$\hat{P}_{i,j} = \frac{1}{P_{i,j}} \tag{9}$$

where, $|\mathcal{D}_i|$ stands for the total number of samples owned by the ith client, and $|\mathcal{D}_{i,k}|$ denotes the total number of samples in category k owned by the ith client. The variable K represents the total number of classes. $\hat{P}_{i,j}$ is the inverse of the distance, which converts the distance into a weight factor.

Moreover, as previous work (McMahan et al., 2017; Oh et al., 2021) has demonstrated, the number of samples each client possesses is a critical factor to consider. Therefore, the formula for calculating the weighting factor from client i to client j is given by:

$$\hat{\alpha}_{i,j} = \mu \cdot \frac{\left|\hat{P}_{i,j}\right|}{\sum_{j=1}^{m} \left|\hat{P}_{i,j}\right|} + (1 - \mu) \cdot \frac{|\mathcal{D}_i|}{\sum_{i=1}^{m} |\mathcal{D}_i|} \tag{10}$$

$$\alpha_{i,j} = \frac{\left|\hat{\alpha}_{i,j}\right|}{\sum_{j=1}^{m} \left|\hat{\alpha}_{i,j}\right|} \tag{11}$$

where μ is a hyper-parameter used to balance the influence of prototype similarity and sample size on the weighting factor. $\alpha_{i,j}$ is obtained by normalizing $\hat{\alpha}_{i,j}$.

4.2.2. Global personalized aggregation for classifier

Unlike the feature extractor, which primarily captures global information, the classifier directly influences the final output and should therefore focus more on learning from local samples. This makes personalized weights more suitable for the classifier. However, to reduce variance, we propose using classifiers from other clients. This approach, while beneficial for variance reduction, may introduce some deviation from the true distribution $P_i(x, y)$. To address this trade-off between deviation and variance, we formulate a quadratic programming problem. The goal is to find an optimal weight distribution for the classifier, balancing these two factors. This is achieved by considering the distance matrix between clients (denoted as matrix P from solving equation (8)) and the intra-class variance δ_i^2, which quantifies the stability of each client's feature distribution. The formulations are given by:

$$p_k = \frac{|\mathcal{D}_{i,k}|}{|\mathcal{D}_i|} \tag{12}$$

$$\delta_i^2 = \frac{1}{|\mathcal{D}_i|} \sum_{k=1}^{K} p_k \left(\frac{\text{Trace}\left(\mathbf{F}_k^\top \mathbf{F}_k\right)}{|\mathcal{D}_{i,k}|} \right) - \sum_{k=1}^{K} p_k^2 \|C_i^{(k)}\|^2 \tag{13}$$

where p_k represents the proportion of class k in the local dataset $|\mathcal{D}_i|$, and $\mathbf{F}_k$ is the feature matrix composed of the hidden vectors of all samples in category k. $C_i^{(k)}$ is also defined as:

$$C_i^{(k)} = \frac{1}{|\mathcal{D}_{i,k}|} \sum_{j=1}^{|\mathcal{D}_{i,k}|} \mathbf{F}_{k,j} \tag{14}$$

The optimization objective of quadratic programming can be described as:

$$\min_{\beta} \frac{1}{2} \beta^\top Q \beta \tag{15}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} \beta_{i,j} = 1, \beta_{i,j} \geq 0 \tag{16}$$

where $Q = \text{Diag}(\delta_i^2) + P$, β is the optimal classifier weight to be solved.

Table 2
Accuracy comparisons of final test accuracy (%) on five benchmark datasets are conducted under Data Heterogeneity Setting 1. We implement full participation in the Federated Learning (FL) system with 20 clients and vary the number of s, 20 and 80, respectively.

Data Heterogeneity Setting 1										
Method	FMNIST		EMNIST		CIFAR-10		CIFAR-100		CINIC-10	
	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$
Local	82.86	77.31	70.6	58.41	55.78	42.93	25.85	18.86	46.81	35.36
FedAvg	86.71	88.05	75.05	73.23	70.01	66.63	44.96	39.71	54.21	54.01
FedAvg-FT	88.65	87.0	79.28	71.06	71.75	70.43	40.13	32.41	56.91	47.78
FedProx	88.05	86.51	77.2	73.53	71.53	69.70	45.53	39.17	57.66	56.40
SCAFFOLD	88.70	87.39	78.13	74.02	72.77	70.13	46.19	40.03	58.3	56.83
FedPer	86.35	83.53	73.79	63.53	61.51	52.41	30.66	21.1	52.01	43.23
LG-FedAvg	83.08	79.18	71.5	60.85	56.36	43.78	29.5	20.61	47.72	38.43
pFedMe	85.72	83.89	73.15	70.92	69.87	62.03	40.12	31.85	54.34	53.67
FedBABU	88.83	87.92	78.22	73.6	73.11	71.22	46.67	38.90	57.81	55.78
pFedAMF	88.12	87.45	80.91	73.15	72.56	71.89	49.23	41.45	56.34	54.12
FedPAC	89.66	88.47	83.12	75.25	76.34	73.07	54.97	45.21	62.62	56.68
FedGPA	**90.04**	**89.15**	**83.45**	**75.96**	**76.73**	**73.33**	**55.79**	**46.08**	**64.40**	**58.77**

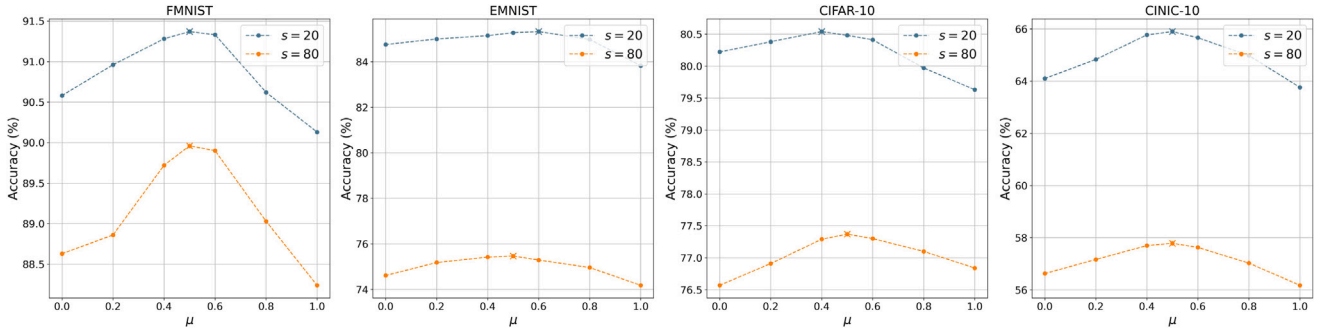

Fig. 3. Tuning of hyper-parameter μ under Data Heterogeneity Setting 3 with $s = 20$ and $s = 80$ on four different datasets.

Table 3
Ablation study. LGA denotes Local–Global Alignment; GPA(f) denotes Global Personalized Aggregation for Feature Extractor; GPA(c) denotes Global Personalized Aggregation for Classifier; +xxx represents adding the current module into the learning procedure.

Dataset	Module usage in FedGPA			
	None	+ LGA	+ GPA(c)	+ GPA(f)
CIFAR-10	70.01	73.18	76.29	**76.73**
FMNIST	86.71	88.54	89.78	**90.04**
EMNIST	75.05	82.13	83.01	**83.45**
CINIC-10	54.21	59.97	63.69	**64.40**
CIFAR-100	44.96	48.64	54.93	**55.79**

5. Experiments

5.1. Experiment setups

5.1.1. Datasets and models

We focus on image classification tasks and evaluate our approach using five benchmark datasets: EMNIST, encompassing 62 classes of handwritten characters; Fashion-MNIST, comprising 10 classes of clothing items; CIFAR-100, consisting of 100 image classes; and CIFAR-10 and CINIC-10, both containing 10 image classes each. We construct two distinct CNN models for EMNIST/Fashion-MNIST and CIFAR-10/CINIC-10/CIFAR-100, respectively. The first CNN model is structured with two convolution layers, each having 16 and 32 channels, followed by a max-pooling layer. It also includes two fully-connected layers with 128 and 10 units, culminating in a softmax output. The second CNN model closely resembles the first, with the addition of one more convolution layer featuring 64 channels.

5.1.2. Federated scenarios

We consider data heterogeneity among clients from three perspectives: the quantity of samples per client (n), the number of dominant classes for each client (k), and the skewness (s) of the class distribution. For each client, $s\%$ of the data (defaulting to 20%) is uniformly sampled from all classes, while the remaining $(100 - s)\%$ is selected from a set of dominant classes (Liu et al., 2023; Xu et al., 2023). Employing these parameters, we simulate heterogeneous settings across three practical scenarios and the data distributions among parties in these settings are shown in Fig. 2.

- **Data Heterogeneity Setting 1.** Each client has an equal number of n and an equal number of dominant classes k. For CIFAR-10 and Fashion-MNIST, n is set to 600, and for CIFAR-100, EMNIST, and CINIC-10, n is set to 1200. k is set to 5 for CIFAR-10, CINIC-10, and Fashion-MNIST; 50 for CIFAR-100; and 31 for EMNIST.
- **Data Heterogeneity Setting 2.** Each client has an equal number of n but a varying number of dominant classes k. For CIFAR-10 and Fashion-MNIST, n is set to 600, and for CIFAR-100, EMNIST, and CINIC-10, n is set to 1200. k is randomly selected from set $A = \{x \in \mathbb{Z} \mid 3 \leq x \leq 7\}$ for CIFAR-10, CINIC-10, and Fashion-MNIST; set $B = \{x \in \mathbb{Z} \mid 30 \leq x \leq 70, x \equiv 0 \pmod 5\}$ for CIFAR-100; and set $C = \{x \in \mathbb{Z} \mid 18 \leq x \leq 42, x \equiv 0 \pmod 3\}$ for EMNIST.
- **Data Heterogeneity Setting 3.** Each client has a different number of n and k. For CIFAR-10 and Fashion-MNIST, n is randomly chosen from $\{300, 900, 1500\}$, and for CIFAR-100, EMNIST, and CINIC-10, n is selected from $\{900, 1500, 2100\}$. k is randomly chosen from set A for CIFAR-10, CINIC-10, and Fashion-MNIST; set B for CIFAR-100; and set C for EMNIST.

5.1.3. Baselines

We compare the following baselines: Local-only, where each client trains its model independently; FedAvg (McMahan et al., 2017) which learns a single global model by aggregating model updates from all clients; FedAvg-FT, a variant of FedAvg where the global model is

Table 4
Accuracy comparisons of final test accuracy (%) on five benchmark datasets are conducted under Data Heterogeneity Setting 2.

Data Heterogeneity Setting 2										
Method	FMNIST		EMNIST		CIFAR-10		CIFAR-100		CINIC-10	
	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$
Local	83.11	78.28	70.61	57.56	56.56	45.01	25.98	20.69	49.01	36.38
FedAvg	88.08	88.14	74.96	72.96	70.56	68.28	42.63	38.33	56.58	54.83
FedAvg-FT	89.16	87.4	78.39	70.56	72.31	66.25	38.81	30.25	58.71	48.16
FedProx	89.0	88.05	79.62	72.77	73.77	70.98	40.31	37.41	58.12	53.99
SCAFFOLD	89.34	88.13	80.01	73.14	73.98	71.57	41.19	37.72	59.26	54.77
FedPer	86.60	84.14	74.31	64.71	62.35	53.56	29.66	23.08	53.31	44.81
LG-FedAvg	84.48	79.67	71.85	63.08	57.18	46.2	27.65	21.88	49.13	36.86
pFedMe	88.12	87.45	78.91	73.15	72.56	71.89	44.23	39.45	57.34	55.12
FedBABU	89.09	88.0	79.03	72.91	74.1	71.76	40.01	36.41	57.43	53.61
pFedAMF	89.45	88.32	80.56	73.35	73.89	71.23	41.56	38.12	59.12	54.95
FedPAC	90.23	88.74	84.32	75.79	76.80	73.55	54.96	47.67	62.63	57.13
FedGPA	**90.41**	**89.26**	**84.63**	**76.50**	**77.41**	**73.95**	**55.25**	**48.12**	**63.01**	**57.45**

first trained collaboratively and then fine-tuned locally for each client to improve personalization; FedProx (Li et al., 2020), an extension of FedAvg that incorporates a proximal term into the local loss function to address data heterogeneity across clients; SCAFFOLD (Karimireddy et al., 2020), which uses control variates to mitigate the drift caused by client data heterogeneity, resulting in more stable updates and faster convergence; LG-FedAvg (Liang et al., 2020), a parameter decoupling approach that separates the global model (shared across all clients) from local models (specific to each client), enabling personalized representation learning; FedPer (Ghuhan Arivazhagan et al., 2019), which globally trains shared base layers while allowing the top layers of the model to be trained locally, balancing generalization and personalization; pFedMe (T. Dinh et al., 2020), which addresses statistical diversity among clients by using Moreau envelopes as regularized loss functions; FedBABU (Oh et al., 2021), which focuses on updating the feature extractor while freezing the classifier during training, allowing each client to train a personalized classifier locally; pFedAMF (Yao et al., 2024), which addresses knowledge forgetting by adaptively fusing global, local, and historical knowledge; and FedPAC (Xu et al., 2023), which combines a shared global feature extractor with a customized classifier for each client to enhance PFL.

5.1.4. Training settings

We employ mini-batch Stochastic Gradient Descent (SGD) as a local optimizer for all approaches, and the number of local training epochs is set to $E = 5$. The local batch size is set to 50, and the learning rate η is set to 0.02. The number of global communication rounds is set to 150, where all approaches have little or no accuracy gain with more communications. We test all methods over three runs and report the average test accuracy across clients.

5.2. Discussions

5.2.1. Model performance

The experimental results under the three data heterogeneity settings are shown in Table 2, Table 4, and Table 5, respectively. It is obvious that our proposed method performs well in any of these three settings. For all datasets, FedGPA dominates the other methods on average test accuracy, which demonstrates the effectiveness and benefit of local–global alignment based on prototype and global personalized aggregation. We also noticed some inspiring experimental results, as follows:

- Under the three data heterogeneity settings, the accuracy of most models is higher when s is smaller than when s is higher.
- When s is smaller, FedAvg with local fine-tuning can serve as a strong baseline for PFL, resulting in competitive performance as a state-of-the-art method. However, when s is larger, FedAvg with local fine-tuning becomes counterproductive.
- The performance of methods like FedAvg, FedProx and SCAFFOLD, which only train a single global model, degrades significantly under these heterogeneity settings since a single global model cannot well accommodate the statistical heterogeneity of clients.
- We notice that traditional PFL methods like FedPer and LG-FedAvg perform poorly compared to recent works like FedBABU and FedPAC. The latter two methods adopt a similar idea: dividing the model into a feature extractor and a classifier, adopting a simple global average strategy for the feature extractor, and employing a personalized weighting strategy for the classifier. Our method builds on this foundation to establish a more effective aggregation method for the feature extractor, resulting in state-of-art accuracy.

5.2.2. Ablation studies

Here, we perform ablation studies to assess the effectiveness of each individually designed module. Without loss of generality, we employ Data Heterogeneity Setting 1 and set s to 20. Specifically, we independently apply Local–Global Alignment (LGA) and Global Personalized Aggregation (GPA) to five benchmark datasets, calculating the average accuracy across 20 clients. As outlined in Table 3, the inclusion of each module results in enhanced average test accuracy, with their combined utilization yielding the most favourable model performance.

5.2.3. Effects of different levels of data heterogeneity

We emulate diverse degrees of data heterogeneity by manipulating parameter s. A value of $s = 0\%$ represents a highly heterogeneous scenario, whereas $s = 80\%$ indicates a greater degree of homogeneity across clients. Our evaluation focuses on the CIFAR-10 dataset under the Data Heterogeneity Setting 1, and the outcomes for various methods are depicted in Fig. 5. Notably, our approach consistently surpasses other baseline methods, underscoring its adaptability and robustness across a variety of heterogeneous data scenarios.

5.2.4. Tuning of hyper-parameter μ

When calculating a client's aggregation weight for the feature extractor, we use the hyper-parameter μ to balance the contributions of prototype-based similarity and sample size in the weighting factor. We tune μ in Eq. (10) across four different datasets: FMNIST, EMNIST, CIFAR-10, and CINIC-10. Without loss of generality, we consider two different values for s, specifically 20 and 80, under Data Heterogeneity Setting 3. In general, μ values between 0.4 and 0.6 provide a good balance, resulting in stable and strong performance. Further details are available in Fig. 3.

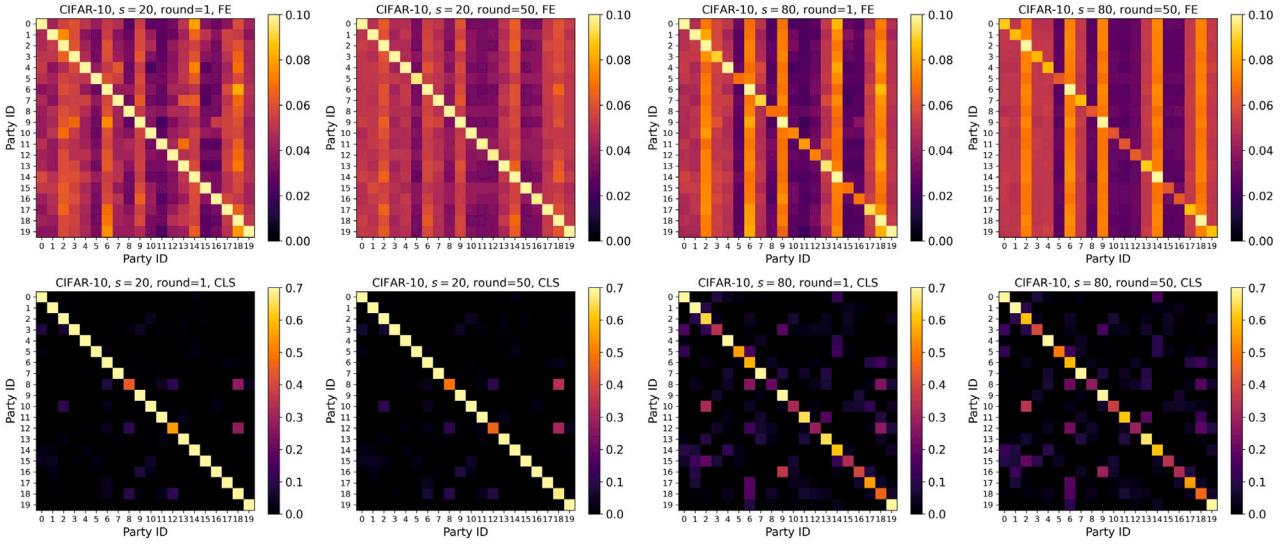

Fig. 4. Visualization of the aggregation weights of the feature extractor (FE) and classifier (CLS) for the CIFAR-10 dataset under Data Heterogeneity Setting 3, with $s = 20$ and $s = 80$.

Table 5
Accuracy comparisons of final test accuracy (%) on five benchmark datasets are conducted under Data Heterogeneity Setting 3.

Data Heterogeneity Setting 3										
Method	FMNIST		EMNIST		CIFAR-10		CIFAR-100		CINIC-10	
	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$
Local	84.96	80.05	74.61	60.2	62.2	48.26	24.86	16.63	49.6	35.46
FedAvg	88.88	86.83	76.91	73.66	73.76	74.87	47.15	43.83	53.66	55.95
FedAvg-FT	90.16	88.56	82.28	71.36	77.01	70.76	44.31	33.65	58.45	48.64
FedProx	89.97	88.73	82.78	74.96	77.83	74.97	44.98	42.93	59.6	55.78
SCAFFOLD	89.68	88.01	83.07	75.14	79.51	75.83	46.03	43.95	60.43	56.73
FedPer	86.93	84.98	78.85	65.51	72.61	65.91	30.46	20.63	56.53	47.34
LG-FedAvg	86.07	82.13	76.13	63.16	66.86	50.93	28.26	20.68	51.2	38.11
pFedMe	87.45	87.12	80.56	73.15	76.23	73.45	43.12	41.56	58.34	54.12
FedBABU	90.0	88.64	82.16	74.21	78.45	75.78	44.85	42.50	59.66	56.25
pFedAMF	89.96	88.34	82.96	74.29	77.89	75.12	46.23	43.12	61.12	55.89
FedPAC	90.70	88.83	85.17	74.91	80.52	76.47	55.75	48.35	64.24	56.85
FedGPA	**91.37**	**89.96**	**85.33**	**75.47**	**80.54**	**77.37**	**56.39**	**48.80**	**65.90**	**57.79**

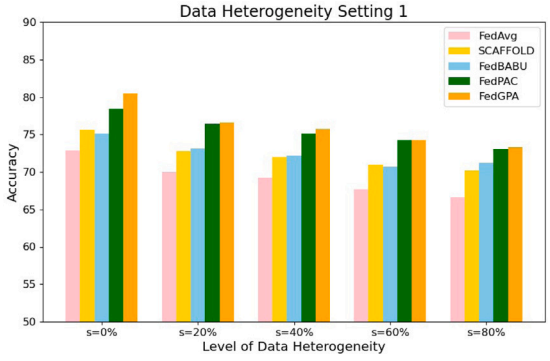

Fig. 5. Model performance on CIFAR-10 dataset with varying s.

5.2.5. Applicability to large or partial client participation scenarios

To evaluate the scalability of FedGPA, we conducted experiments with varying numbers of clients. Without loss of generality, we employ Data Heterogeneity Setting 3 under CIFAR-10 dataset. Following the methodology of FedProc (Mu et al., 2023), the number of clients (m) is set to 20 (with sampling rate $\gamma = 1$), 50 (with sampling rate $\gamma = 1$), and 100 (with sampling rate $\gamma = 0.2$). Note that $\gamma = 0.2$ means that 20 clients are randomly selected from a total of 100 to participate in training during each communication round (for more on client sampling techniques, see FedAvg (McMahan et al., 2017)). The results of these experiments are presented in Table 6.

Table 6
Performance of various methods under different client participation scenarios on CIFAR-10 in the context of Data Heterogeneity Setting 3.

Method	$m = 20$	$m = 50$	$m = 100$
	$\gamma = 1$	$\gamma = 1$	$\gamma = 0.2$
FedAvg	73.76	73.84	69.62
SCAFFOLD	79.51	80.03	77.35
FedBABU	78.45	78.0	77.61
FedPAC	80.52	80.94	79.72
FedGPA	**80.54**	**81.28**	**80.13**

5.2.6. Visualization of the weight distribution of α and β in federated aggregation

We detailed the design rationale and calculation methods for the weighting factors, α and β, used in the personalized aggregation of feature extractors (FE) and classifiers (CLS) in Section 4.2. To illustrate this, we present a heatmap visualization of the aggregation strategy, as shown in Fig. 4. Using the CIFAR-10 dataset, we plot the weight matrices for the FE and CLS under different conditions: $s = 20$ and 80, and round = 1, 50. Key observations include: (i) Compared to classifiers,

Table 7
The average training time per communication round.

Method	FMNIST	CIFAR-10	CIFAR-100
FedAvg	6 min 10 s	7 min 2 s	22 min 16 s
FedProx	6 min 24 s	7 min 36 s	24 min 18 s
SCAFFOLD	6 min 50 s	8 min 7 s	23 min 47 s
FedBABU	6 min 23 s	7 min 12 s	23 min 6 s
FedPAC	7 min 28 s	8 min 43 s	26 min 3 s
FedGPA	7 min 33 s	8 min 57 s	26 min 17 s

feature extractors exhibit a more evenly distributed weight aggregation, promoting generalization. (ii) As data heterogeneity increases (with s decreasing from 80 to 20), both classifiers and feature extractors tend to assign more weight to their own local model, with classifiers being more influenced due to the design of intra-class variance.

5.2.7. Communication and computation efficiency

Suppose that there are m clients participating in the training process, the number of model parameters is $|w|$, the number of classes in the dataset is k, and the dimension of the vector output by the feature extractor is d. The complexity of our purposed FedGPA algorithm is $\mathcal{O}\left(m^2|w| + mkd\right)$. The first term represents the overhead caused by the server receiving and transmitting the model parameters of each client, while the second term represents the overhead from the server receiving the prototype matrix from each client and calculating the similarity. Generally, the product of the number of classes and the dimension of the hidden layer vector is much smaller than the number of model parameters $|w|$, so the overall time complexity is approximately to $\mathcal{O}\left(m^2|w|\right)$.

To make it more intuitive, we evaluate the computational cost of several methods discussed in our experiments using the same hardware. Table 7 presents the average training time per communication round. It is evident that FedAvg has the lowest average training time among all the methods. This is because FedProx introduces additional loss terms based on FedAvg, while SCAFFOLD adds extra control variables for both the server and the clients. FedBABU, FedPAC, and FedGPA generate personalized models for each client, which incurs additional computational effort. Specifically, FedBABU requires further model fine-tuning on the client side, and FedPAC generates personalized classifiers by solving optimization problems. We observed that FedGPA slightly increased the computational cost compared to FedPAC, as the former additionally considers personalized aggregation of feature extractors.

5.2.8. Privacy

Compared with traditional FL methods, such as FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), SCAFFOLD (Karimireddy et al., 2020), etc., our proposed FedGPA framework requires the exchange of prototypes between the server and the clients. However, this does not bring the risk of data leakage for these reasons. Firstly, prototypes inherently protect data privacy, as they are 1D vectors generated by averaging the low-dimensional representations of samples from the same class, a process that is irreversible. Secondly, attackers cannot reconstruct the raw data from prototypes without access to the local models. These indicate that this process is more privacy-preserving compared with several existing works, such as FedFTG (Zhang et al., 2022), which transmitting exact category distribution. Moreover, FedGPA can be combined with various privacy-preserving techniques to further strengthen the system's security and reliability.

6. Conclusion

In this paper, we propose a novel FedGPA algorithm. Comprehensive experiments on four datasets demonstrate the superb performance of FedGPA which can achieve the highest model accuracy compared to the state-of-the-art baselines under three practical heterogeneous FL settings. Although effective, FedGPA may increase the communication cost of federated learning. In our future work we will explore to handle this problem.

CRediT authorship contribution statement

Zongfu Han: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Yu Feng:** Methodology, Formal analysis, Conceptualization. **Yifan Zhu:** Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Zhen Tian:** Supervision, Resources, Conceptualization. **Fangyu Hao:** Software, Data curation. **Meina Song:** Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 62406036, the National Key Research and Development Program of China under Grant 2024YFC3308503, the Key Laboratory of Target Cognition and Application Technology under Grant 2023-CXPT-LC-005, and also sponsored by SMP-Zhipu.AI Large Model Cross-Disciplinary Fund under Grant ZPCG20241029322.

Appendix. Additional experimental results

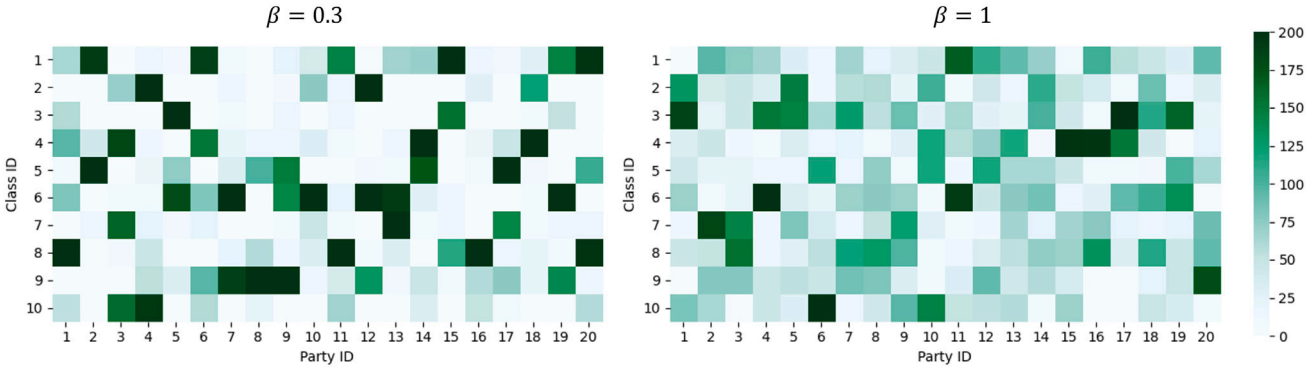

Fig. 6. Data distribution across parties under Data Heterogeneity Setting 1 (taking CIFAR-10 as an example) using the Dirichlet distribution.

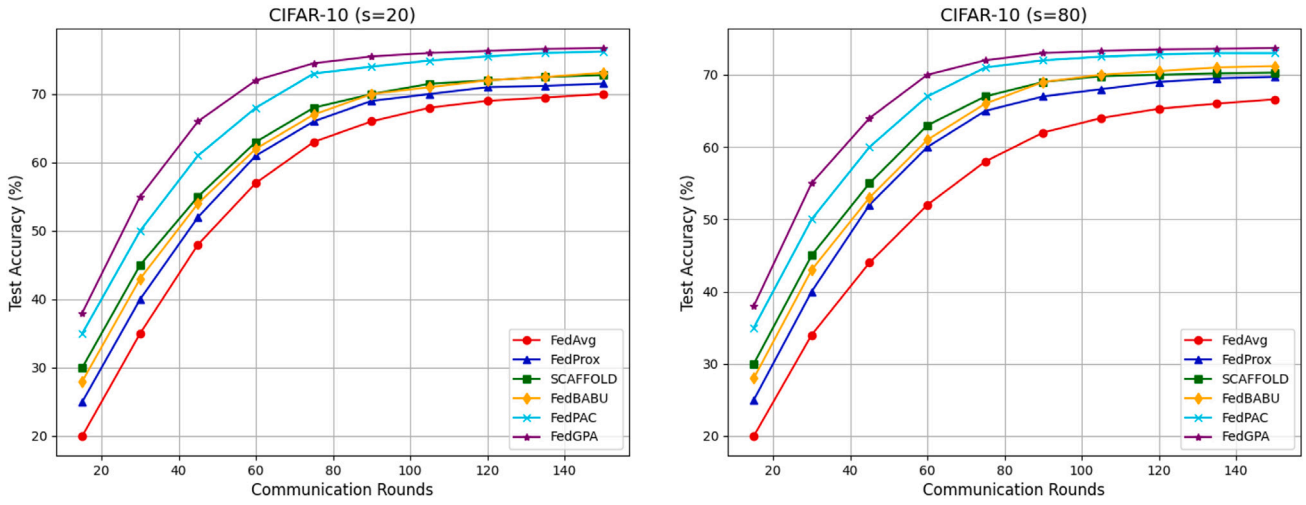

Fig. 7. Data distribution across parties under Data Heterogeneity Setting 1 (taking CIFAR-10 as an example) using the Dirichlet distribution.

Table A.8
Accuracy comparisons of final test accuracy (%) on five benchmark datasets are conducted under Data Heterogeneity Setting 1*.

Data Heterogeneity Setting 1*										
Method	FMNIST		EMNIST		CIFAR-10		CIFAR-100		CINIC-10	
	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$	$s = 20$	$s = 80$
Local	83.05	77.76	71.27	59.72	56.84	44.16	26.04	19.93	47.52	35.90
FedAvg	87.52	88.70	76.54	73.78	70.86	67.94	45.21	39.97	55.09	54.41
FedAvg-FT	89.91	87.48	80.63	72.64	72.68	71.67	40.71	32.91	57.56	48.60
FedProx	88.68	86.86	78.02	74.12	72.25	70.85	45.89	39.60	58.49	56.92
SCAFFOLD	89.12	87.83	78.93	74.70	73.10	70.86	46.62	40.32	58.92	57.31
FedPer	87.23	84.12	74.11	64.32	62.03	53.12	31.56	21.38	52.79	43.97
LG-FedAvg	83.73	79.91	72.03	61.11	57.25	44.57	29.73	20.80	48.19	38.86
FedBABU	89.48	88.56	78.86	74.32	73.45	71.61	47.13	39.14	58.08	56.20
FedPAC	90.02	88.97	83.85	75.83	77.12	73.72	55.13	45.77	63.12	57.42
FedGPA	**90.48**	**89.17**	**83.96**	**76.47**	**77.34**	**73.85**	**55.98**	**46.09**	**64.16**	**58.03**

Table A.9
Accuracy comparisons of final test accuracy (%) on five benchmark datasets are conducted under Data Heterogeneity Setting 1 (Dirichlet).

Data Heterogeneity Setting 1 (Dirichlet)										
Method	FMNIST		EMNIST		CIFAR-10		CIFAR-100		CINIC-10	
	$Dir(0.3)$	$Dir(1)$	$Dir(0.3)$	$Dir(1)$	$Dir(0.3)$	$Dir(1)$	$Dir(0.3)$	$Dir(1)$	$Dir(0.3)$	$Dir(1)$
Local	83.89	78.12	72.11	60.12	57.64	44.89	27.05	20.78	46.35	35.31
FedAvg	88.32	89.13	77.97	74.58	70.16	67.12	45.89	40.34	54.52	54.10
FedAvg-FT	89.87	88.88	80.32	73.49	72.89	71.43	41.40	33.11	57.22	48.34
FedProx	88.97	87.03	78.34	74.46	72.98	71.47	46.12	39.20	58.72	56.83
SCAFFOLD	89.14	88.11	78.79	74.44	73.56	71.12	46.81	40.85	58.39	57.63
FedPer	87.77	84.67	74.69	64.94	62.71	53.85	31.45	21.12	52.90	44.24
LG-FedAvg	83.50	80.43	72.61	61.81	58.06	44.21	29.87	20.16	47.88	39.48
pFedMe	86.26	84.06	73.83	70.48	70.04	61.68	40.29	32.72	54.84	53.15
FedBABU	89.91	88.91	79.34	74.77	73.58	71.98	47.79	39.41	58.56	56.51
pFedAMF	88.60	87.88	81.17	73.75	72.97	72.56	49.80	41.82	56.55	54.22
FedPAC	90.47	89.35	84.32	76.09	77.23	73.59	55.63	45.85	63.68	57.65
FedGPA	**90.91**	**89.55**	**84.61**	**77.01**	**77.73**	**74.13**	**56.25**	**46.36**	**64.85**	**58.87**

A.1. Scalability with increasing data size

To more thoroughly evaluate the effectiveness of our proposed FedGPA, we increase the number of samples for each client under Data Heterogeneity Setting 1, which we refer to as Data Heterogeneity Setting 1*. For the CIFAR-10 and Fashion-MNIST datasets, each client is assigned 2000 samples (increased from 600), while for the CIFAR-100, EMNIST, and CINIC-10 datasets, each client is assigned 4000 samples (increased from 1200). The corresponding experimental results are presented in Table A.8. These results demonstrate that FedGPA continues to deliver the best performance as the number of samples increases.

A.2. More experiments under Dirichlet distribution

In our design to simulate heterogeneous environments, we primarily consider three factors: the number of samples per client n, the number of dominant classes per client k, and the skewness s of the class distribution. For each client, $s\%$ of the data (defaulting to 20%) is uniformly sampled from all classes, while the remaining $100 - s\%$

is selected from a set of dominant classes. This design aligns with methodologies employed in prior studies, such as FedPAC (Xu et al., 2023) and FedDWA (Liu et al., 2023). However, it is important to note that another widely adopted approach involves modelling data heterogeneity using the Dirichlet distribution function (Mu et al., 2023; Yao et al., 2024).

Consequently, we partition the client dataset using the Dirichlet distribution, following the data heterogeneity scenario outlined in Setting 1. The hyper-parameter β is set to 0.3 and 1, simulating different levels of data heterogeneity. In this setting, the total number of samples across all clients is 600. We use the Dirichlet distribution to partition the dataset, and the number of categories assigned to each client is shown in Fig. 6. The detailed experimental results can be found in Table A.9 . The results demonstrate that our method continues to achieve the best performance, which, to some extent, highlights the superiority and generalization ability of our approach.

A.3. Curves of test accuracy

Fig. 7 presents the evolution of average test accuracy over global communication rounds for partial experiments shown in Table 2. From which, it can be seen that our method almost outperforms other baselines during the whole training process. The results confirm that our method not only converges faster but also achieves better performance under the same computational and communication conditions.

References

Adnan, M., Kalra, S., Cresswell, J.C., Taylor, G.W., Tizhoosh, H.R., 2022. Federated learning and differential privacy for medical image analysis. Sci. Rep. 12 (1), 1953.

Andreina, S., Marson, G.A., Möllering, H., Karame, G., 2021. Baffle: Backdoor detection via feedback-based federated learning. In: 2021 IEEE 41st International Conference on Distributed Computing Systems. ICDCS, IEEE, pp. 852–863.

Aono, Y., Hayashi, T., Wang, L., Moriai, S., et al., 2017. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans. Inf. Forensics Secur. 13 (5), 1333–1345.

Asad, M., Moustafa, A., Ito, T., 2020. Fedopt: Towards communication efficiency and privacy preservation in federated learning. Appl. Sci. 10 (8), 2864.

Bietti, A., Wei, C.-Y., Dudik, M., Langford, J., Wu, S., 2022. Personalization improves privacy-accuracy tradeoffs in federated learning. In: International Conference on Machine Learning. PMLR, pp. 1945–1962.

Briggs, C., Fan, Z., Andras, P., 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In: 2020 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1–9.

Chen, H.-Y., Chao, W.-L., 2021. On bridging generic and personalized federated learning for image classification. arXiv preprint arXiv:2107.00778.

Chen, X., Du, T., Wang, M., Gu, T., Zhao, Y., Kou, G., Xu, C., Wu, D.O., 2024. Towards optimal customized architecture for heterogeneous federated learning with contrastive cloud-edge model decoupling. IEEE Trans. Comput..

Collins, L., Hassani, H., Mokhtari, A., Shakkottai, S., 2021. Exploiting shared representations for personalized federated learning. In: International Conference on Machine Learning. PMLR, pp. 2089–2099.

Dai, R., Shen, L., He, F., Tian, X., Tao, D., 2022. Dispfl: Towards communication-efficient personalized federated learning via decentralized sparse training. arXiv preprint arXiv:2206.00187.

Feng, J., Wu, Y., Sun, H., Zhang, S., Liu, D., 2025. Panther: Practical secure 2-party neural network inference. IEEE Trans. Inf. Forensics Secur..

Fu, L., Zhang, H., Gao, G., Zhang, M., Liu, X., 2023. Client selection in federated learning: Principles, challenges, and opportunities. IEEE Internet Things J. 10 (24), 21811–21819.

Ghosh, A., Chung, J., Yin, D., Ramchandran, K., 2020. An efficient framework for clustered federated learning. Adv. Neural Inf. Process. Syst. 33, 19586–19597.

Ghuhan Arivazhagan, M., Aggarwal, V., Singh, A.K., Choudhary, S., 2019. Federated learning with personalization layers. arXiv e-prints, arXiv–1912.

Guo, X., Yu, H., Li, B., Wang, H., Xing, P., Feng, S., Nie, Z., Miao, C., 2022. Federated learning for personalized humor recognition. ACM Trans. Intell. Syst. Technol. (TIST) 13 (4), 1–18.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., Ramage, D., 2018. Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604.

Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al., 2021. Advances and open problems in federated learning. Found. Trends® Mach. Learn. 14 (1–2), 1–210.

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

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Lee, R., Kim, M., Li, D., Qiu, X., Hospedales, T., Huszár, F., Lane, N., 2023. Fedl2p: Federated learning to personalize. Adv. Neural Inf. Process. Syst. 36, 14818–14836.

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

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V., 2020. Federated optimization in heterogeneous networks. Proc. Mach. Learn. Syst. 2, 429–450.

Liang, P.P., Liu, T., Ziyin, L., Allen, N.B., Auerbach, R.P., Brent, D., Salakhutdinov, R., Morency, L.-P., 2020. Think locally, act globally: Federated learning with local and global representations. arXiv preprint arXiv:2001.01523.

Lin, T., Kong, L., Stich, S.U., Jaggi, M., 2020. Ensemble distillation for robust model fusion in federated learning. Adv. Neural Inf. Process. Syst. 33, 2351–2363.

Liu, Y., Huang, A., Luo, Y., Huang, H., Liu, Y., Chen, Y., Feng, L., Chen, T., Yu, H., Yang, Q., 2020. Fedvision: An online visual object detection platform powered by federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 13172–13179.

Liu, J., Wu, J., Chen, J., Hu, M., Zhou, Y., Wu, D., 2023. FedDWA: Personalized federated learning with online weight adjustment. arXiv preprint arXiv:2305.06124.

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

Michieli, U., Ozay, M., 2021. Prototype guided federated learning of visual feature representations. arXiv preprint arXiv:2105.08982.

Mu, X., Shen, Y., Cheng, K., Geng, X., Fu, J., Zhang, T., Zhang, Z., 2023. Fedproc: Prototypical contrastive federated learning on non-iid data. Future Gener. Comput. Syst. 143, 93–104.

Oh, J., Kim, S., Yun, S.-Y., 2021. Fedbabu: Towards enhanced representation for federated image classification. arXiv preprint arXiv:2106.06042.

Ruan, Y., Joe-Wong, C., 2022. Fedsoft: Soft clustered federated learning with proximal local updating. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 8124–8131.

Snell, J., Swersky, K., Zemel, R., 2017. Prototypical networks for few-shot learning. Adv. Neural Inf. Process. Syst. 30.

T. Dinh, C., Tran, N., Nguyen, J., 2020. Personalized federated learning with moreau envelopes. Adv. Neural Inf. Process. Syst. 33, 21394–21405.

Tan, A.Z., Yu, H., Cui, L., Yang, Q., 2022. Towards personalized federated learning. IEEE Trans. Neural Netw. Learn. Syst. 34 (12), 9587–9603.

Xu, J., Tong, X., Huang, S.-L., 2023. Personalized federated learning with feature alignment and classifier collaboration. arXiv preprint arXiv:2306.11867.

Yao, D., Zhu, Z., Liu, T., Xu, Z., Jin, H., 2024. Rethinking personalized federated learning from knowledge perspective. In: Proceedings of the 53rd International Conference on Parallel Processing. pp. 991–1000.

Zhang, L., Shen, L., Ding, L., Tao, D., Duan, L.-Y., 2022. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10174–10183.