Personalized Federated Learning with Inferred Collaboration Graphs

Rui Ye [*1] Zhenyang Ni [*1] Fangzhao Wu [2] Siheng Chen [1 3] Yanfeng Wang [3 1]

Abstract

Personalized federated learning (FL) aims to collaboratively train a personalized model for each client. Previous methods do not adaptively determine who to collaborate at a fine-grained level, making them difficult to handle diverse data heterogeneity levels and those cases where malicious clients exist. To address this issue, our core idea is to learn a collaboration graph, which models the benefits from each pairwise collaboration and allocates appropriate collaboration strengths. Based on this, we propose a novel personalized FL algorithm, pFedGraph, which consists of two key modules: (1) inferring the collaboration graph based on pairwise model similarity and dataset size at server to promote fine-grained collaboration and (2) optimizing local model with the assistance of aggregated model at client to promote personalization. The advantage of pFedGraph is flexibly adaptive to diverse data heterogeneity levels and model poisoning attacks, as the proposed collaboration graph always pushes each client to collaborate more with similar and beneficial clients. Extensive experiments show that pFedGraph consistently outperforms the other 14 baseline methods across various heterogeneity levels and multiple cases where malicious clients exist. Code will be available at https://github.com/MediaBrain-SJTU/pFedGraph.

1. Introduction

As an emerging collaborative machine learning framework with privacy-preserving properties, federated learning (FL) (McMahan et al., 2017) has attracted much attention from both industries (Yang et al., 2018; Li et al., 2020a) and academia (Yang et al., 2019; Wang et al., 2021).

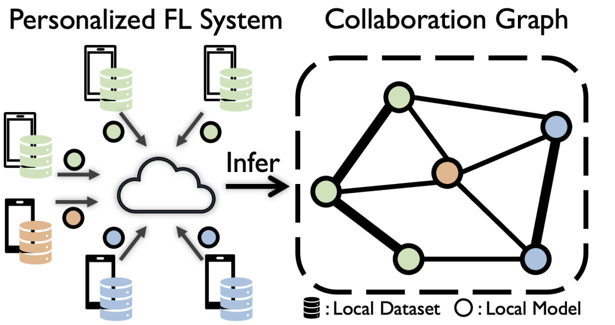


Figure 1. Inferred collaboration graph guides the server to coordinate which pairs of clients should collaborate at what intensity level, enabling the proposed pFedGraph to be adaptive to diverse data heterogeneity levels.

In the conventional paradigm, multiple clients collaboratively train a shared global model without transmitting local datasets (McMahan et al., 2017). However, due to data heterogeneity across clients, the global model trained with all data from all clients might not cater to the utility of each individual client. To promote better customization to each client, personalized FL has been proposed and allows each client to train a personalized model that is adapted to its local data (Smith et al., 2017; Li et al., 2021b).

A fundamental challenge in personalized FL is an appropriate tradeoff between the individual utilities and the benefits through collaboration. When all clients have similar data distributions, they should get much improvements through collaborating with each other; while when each client's data distribution is different, collaboration would bring less benefits to each client. Therefore, the optimal individual and collaboration tradeoff depends on data heterogeneity level. However, in practice, data heterogeneity level is unknown since data is not communicated in FL, which makes this tradeoff hard to handle. To handle this issue, previous works have put forth solutions from several perspectives. For example, Ditto (Li et al., 2021b) and pFedMe (T Dinh et al., 2020) regularizes the ℓ_2 distance between the local and global models. L2GD (Hanzely & Richtárik, 2020) and APFL (Deng et al., 2020) consider linear interpolation between local and global models. FedPer (Arivazhagan et al., 2019), FedRep (Collins et al., 2021), and FedRoD (Chen & Chao, 2021) share the same aggregated shallow model

[*] Equal contribution [1] Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China [2] Microsoft Research Asia, Beijing, China [3] Shanghai AI Laboratory, Shanghai, China. Correspondence to: Siheng Chen <sihengc@sjtu.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

layers and hold the localized deep layer. CFL (Sattler et al., 2020) proposes clustering and FedAMP (Huang et al., 2021) proposes neighbor weights optimizing that rely on manual tuning to adjust collaboration strategy.

However, all previous methods do not specifically determine which pairs of clients should collaborate at what intensity level. This weakness makes those methods less flexible to handle diverse data heterogeneity levels at various clients, and cases when malicious clients exist. To address this issue, our core idea is to learn a collaboration graph, revealing who to collaborate for each client at each communication round. In this graph, each node denotes the personalized model of a client and each edge weight reflects the collaboration intensity between two personalized models and is updated at each communication round. Intuitively, when two clients have similar data distributions, their personalized models tend to be similar and there would be more benefits for them to collaborate. Thus, the pairwise connectivity in this collaboration graph could serve as a proxy to reflect the benefits brought by each collaboration.

With this graph-based design rationale, we formulate a novel optimization problem for personalized FL, which simultaneously optimizes personalized models and the collaboration graph. Our formulation naturally captures the fundamental tradeoff between the individual utilities and the collaboration benefits modeled by the proposed collaboration graph. To solve this optimization problem, we propose a novel personalized FL algorithm, pFedGraph, which consists of two main steps: inferring a collaboration graph at the server side and optimizing personalized models at the client side. The server learns a collaboration graph by promoting the correlation with model similarity, and then, obtains an aggregated model for each client according to the learned collaboration graph. At the client side, a personalized model is optimized through balancing between empirical task-driven loss and the similarity between the local personalized model and the aggregated model sent from the server.

The advantage of the proposed pFedGraph is adaptivity to diverse data heterogeneity levels, because the inferred collaboration graph can determine who to collaborate and push each client to work with other clients with similar data distributions, maintaining local data homogeneity. Specifically, compared to Ditto (Li et al., 2021b) and pFedMe (T Dinh et al., 2020), which regularizes personalized models to be close to the shared global model, the proposed pFedGraph can distinguish beneficial collaborators from irrelevant or malicious users, avoiding blind collaboration. Compared to CFL (Sattler et al., 2020), which clusters similar clients into groups, pFedGraph can reflect each pairwise collaboration relationship, which is much more fine-grained.

We compare our proposed pFedGraph with 14 representative baselines on diverse tasks (including image and text classification), datasets (Fashion-MNIST (Xiao et al., 2017), CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009) and Yahoo! Answers (Zhang et al., 2015)) and scenarios (different heterogeneous levels and model poisoning settings). The results show that pFedGraph is robust towards i) different levels of data heterogeneity and ii) model poisoning attack compared to these existing personalized FL methods.

Our contribution is summarized as follows:

• We formulate a novel personalized FL optimization problem, which tackles the tradeoff between the individual utilities and the collaboration benefits modeled by the proposed collaboration graph.

• We propose pFedGraph algorithm, which infers the collaboration graph at the server side and optimizes personalized models at the client side. pFedGraph is adaptive to diverse data heterogeneity levels.

• We conduct extensive experiments to show pFedGraph outperforms the other 14 personalized FL methods across various data heterogeneity levels as well as multiple model poisoning attack settings.

2. Related Work

In this section, we discuss related work from the perspectives of both general federated learning and personalized federated learning. We also provide more detailed comparisons with several methods in Appendix B.

2.1. Federated Learning

The goal of federated learning is to enable collaborative training of machine learning models at multiple clients in a privacy-preserving manner. As the pioneering work, FedAvg (McMahan et al., 2017) is the template of most existing FL methods. After that, FedProx (Li et al., 2020b) and FedDyn (Acar et al., 2020) focus on model-level regularization by restricting the ℓ_2 distance between each local model and global model. MOON (Li et al., 2021a) conducts feature alignment between local and global model while FedFM (Ye et al., 2022) aligns category-wise feature spaces across clients. SCAFFOLD (Karimireddy et al., 2020) introduces control variate to correct the gradient of local model. FedNova (Wang et al., 2020b) modifies model aggregation weights according to number of local updates. FedAvgM (Hsu et al., 2019) and FedOPT (Reddi et al., 2021) introduce server-side momentum to stabilize global model updating, FedDF (Lin et al., 2020) distills knowledge from local models to the global model based on public dataset. However, those federated learning methods focus on training one global model that could perform well on the average of all the clients, and neglect the local demand at each individual client.

2.2. Personalized Federated Learning

Personalized federated learning aims to improve the performance of each client's model through collaborative training. Optimization-based methods such as Ditto (Li et al., 2021b) and pFedMe (T Dinh et al., 2020) propose to restrict the ℓ_2 difference between personalized model and global model. Aggregation-based methods modify the aggregation manner, where CFL (Sattler et al., 2020) divides clients into several clusters of federated learning process based on gradient similarity, FedAMP (Huang et al., 2021) adjusts neighbors weights based on parameters difference with a pre-defined self weight and KT-pFL (Zhang et al., 2021) relies on public dataset to measure the similarity of features extracted by local models. Methods such as Per-FedAvg (Fallah et al., 2020) consider using meta-learning (MAML) to learn a better initial shared model for each client (Jiang et al., 2019). FedPer (Arivazhagan et al., 2019), LG-FEDAVG (Liang et al., 2020) and FedRep (Collins et al., 2021) consider splitting model layers which can be either personalized or shared, e.g., clients share the same feature extractor and train a personalized fully-connected layer locally in FedRep. pFedHN (Shamsian et al., 2021) and FedRoD (Chen & Chao, 2021) train a hypernetwork to produce a personalized fully-connected layer given client's category distribution and KNN-Per (Marfoq et al., 2022) memorizes local prototypes to assist personalized prediction, though they are limited to classification task.

Compared to those previous personalized FL methods, the distinct advantage of the proposed pFedGraph is the adaptation to diverse data heterogeneity levels. In pFedGraph, a collaboration graph is learnt to explicitly specify who to collaborate at what intensity level at each communication round. In this manner, pFedGraph allows each client to work more with similar clients, addressing the diverse data heterogeneity issue.

3. Methodology

In this section, we first formulate a novel personalized FL optimization problem, which aims to optimize the personalized models and the collaboration graph to achieve better tradeoff between the individual utilities and the benefits through collaboration. Then, we decompose this problem into the server and client sides, according with practical architecture constraints. Finally, we propose the pFedGraph (personalized federated learning with collaboration graph) algorithm to solve this optimization problem.

3.1. Optimization Problem

Assume there are K clients. For the ith client c_i, let $\mathcal{D}_i$ be the local dataset with $n_i = |\mathcal{D}_i|$ be the dataset size, $\boldsymbol{\theta}_i$ be the local personalized model and $F_i(\boldsymbol{\theta}_i) = \sum_{\xi \in \mathcal{D}_i} \ell(\boldsymbol{\theta}_i, \xi)/n_i$ be the local empirical loss with $\ell(\cdot,\cdot)$ the pre-defined task-driven loss function. To model the collaboration relationships between clients, we consider a collaboration graph $\mathcal{G}(\mathcal{V}, \boldsymbol{W})$, where the node set $\mathcal{V} = \{c_1, c_2, \cdots, c_K\}$ is a collection of all K clients and $\boldsymbol{W} \in \mathbb{R}^{K \times K}$ is the graph adjacency matrix whose (i,j)th element reflects the collaboration relationship between the ith and the jth clients. Intuitively, when two clients have more similar data distributions, their corresponding models should be more similar and their collaboration strength should be larger to allow these two clients to learn more from each other (Li et al., 2020b; Luo et al., 2021). In practice, data distributions are inaccessible, we use the similarity between model parameters to guide the collaboration strength. Then, the proposed optimization problem for personalized FL is

$$\min_{\{\boldsymbol{\theta}_i\},\boldsymbol{W}} \sum_{i=1}^{K} p_i \left(F_i(\sum_{j=1}^{K} \boldsymbol{W}_{ij}\boldsymbol{\theta}_j) - \frac{\lambda}{2} \sum_{j=1}^{K} \boldsymbol{W}_{ij} \cos(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \right)$$
$$s.t. \ \sum_{j=1}^{K} \boldsymbol{W}_{ij} = 1, \forall i; \quad \boldsymbol{W}_{ij} \geq 0, \forall i, j, \qquad (1)$$

where $p_i = n_i / \sum_j n_j$ is the relative dataset size, λ is a hyperparameter to balance the individual utilities and the collaboration necessity, and $\cos(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \left\langle \frac{\boldsymbol{\theta}_i}{||\boldsymbol{\theta}_i||}, \frac{\boldsymbol{\theta}_j}{||\boldsymbol{\theta}_j||} \right\rangle$ is the cosine similarity between the ith and the jth client's personalized models before collaboration. Problem (1) optimizes both personalized models and the collaboration graph. In the objective function, the first term models the empirical loss at each client after collaboration, where $\widetilde{\boldsymbol{\theta}}_i = \sum_j \boldsymbol{W}_{ij}\boldsymbol{\theta}_j$ is the collaborated (aggregated) model at the ith client. The second term measures the correlation between collaboration strength and the model similarity, guiding each client to collaborate more with those others whose model parameters are similar. The first constraint limits the overall collaboration budge at each client and the second one requires all the collaboration strengths be non-negative.

Compared to many previous works, this formulation is novel from two aspects. First, we use a collaboration graph to explicitly reflect who to collaborate at what intensity level, eliminating unrelated or malicious clients. Therefore, each client only needs to collaborate with similar clients, promoting local data homogeneity and leading to stronger adaption ability to handle diverse data heterogeneity levels. Second, the model similarity is measured by the cosine similarity, while most previous works use ℓ_2 distance. The benefits are that cosine similarity 1) better reflects similarity (also see Figure 4(a)) as it is not sensitive towards absolute magnitude of model parameters and 2) naturally normalizes the similarity to a fixed range, which eases hyper-parameter tuning under different scenarios.

This problem can be solved by iteratively solving $\{\boldsymbol{\theta}_i\}$ and

$\boldsymbol{W}$ in a centralized manner, i.e., both datasets and local models are accessible by the central server. However, the practical FL architecture setting causes two constraints: i) the server can only access the local models $\{\boldsymbol{\theta}_i\}$ uploaded from clients but cannot access the datasets to compute the loss value $\{F_i(\cdot)\}$; and ii) each client can only access its own dataset to compute its loss value $F_i(\cdot)$ and an aggregated model from the server but cannot access each individual local model $\{\boldsymbol{\theta}_j\}_{j\neq i}$.

3.2. Problem Decomposition in FL architecture

To handle these constraints, we decompose the original problem (1) into two parts: inferring the collaboration graph $\boldsymbol{W}$ at the server side and optimizing personalzied models $\{\boldsymbol{\theta}_i\}$ at the client side.

Solving $\boldsymbol{W}$ at the server side. As the server cannot access the local datasets to evaluate clients' loss value $\{F_i(\cdot)\}$, we can only approximate the first term in Equation (1). Without knowing any other information about clients, we assume that clients with larger dataset sizes are more reliable collaborators. We thus promote collaboration among clients by regularizing the collaboration strength to align with the relative dataset size. Then, the optimization for the ith client at the server side is:

$$\begin{aligned}\min_{\{\boldsymbol{W}_{ij}\}_j} & \sum_j (\boldsymbol{W}_{ij} - p_j)^2 - \alpha \sum_j \boldsymbol{W}_{ij} \cos(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \\ s.t. & \sum_j \boldsymbol{W}_{ij} = 1, \forall i; \quad \boldsymbol{W}_{ij} \geq 0, \forall i, j, \end{aligned} \tag{2}$$

where $p_j = n_j / \sum_{j'} n_{j'}$ and α is a hyper-parameter. Here the first term promotes a dataset-size-weighted collaboration, as the model trained by more data samples tends to carry more knowledge and deserve a higher collaboration strength. The second term is directly inherited from (1). More explanations of such approximation are in Appendix A.1.

Solving $\boldsymbol{\theta}_i$ at the client side. Without loss of generality, the optimization for the ith client is:

$$\begin{aligned}\min_{\boldsymbol{\theta}_i} H_i(\boldsymbol{\theta}_i) &= F_i(\sum_j \boldsymbol{W}_{ij}\boldsymbol{\theta}_j) - \frac{\lambda}{2}\sum_j \boldsymbol{W}_{ij}\cos(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \\ &= F_i(\sum_j \boldsymbol{W}_{ij}\boldsymbol{\theta}_j) - \frac{\lambda}{2}\left\langle \frac{\boldsymbol{\theta}_i}{||\boldsymbol{\theta}_i||}, \sum_j \boldsymbol{W}_{ij}\frac{\boldsymbol{\theta}_j}{||\boldsymbol{\theta}_j||}\right\rangle,\end{aligned} \tag{3}$$

where the first term minimizes empirical task-driven loss to pursue local model utility. The second term maximizes the model cosine similarity between the local model $\boldsymbol{\theta}_i$ and the aggregated model sent from the server, which avoids drifting too far from the aggregated model and over-fitting to the local dataset.

3.3. pFedGraph Algorithm

Based on the above optimization, we propose a personalized federated learning algorithm with inferred collaboration graphs (pFedGraph, Algorithm 1), which also consists of two major FL procedures as conventional FL method (McMahan et al., 2017): i) model aggregating and broadcasting at the server side; ii) model training and uploading at the client side. Our modifications focus on i) optimizing collaboration graph for model aggregating and ii) optimizing local model with regularization during model training. Also see overview of pFedGraph in Figure 5.

Optimizing collaboration graph at the server side. Based on the local personalized models $\{\boldsymbol{\theta}_j\}_j$ uploaded from the clients, the server first computes their pair-wise cosine similarity of model parameters, that is, each element in the similarity matrix $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ denotes the similarity between the model parameters of the i and j clients: $\boldsymbol{S}_{ij} = \cos(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$. We can rewrite the optimization problem for each client i in (2) as a standard quadratic program:

$$\begin{aligned}\boldsymbol{w}_i = \quad & \arg\min_{\boldsymbol{x}} \quad \boldsymbol{x}^T\boldsymbol{x} + (-2\boldsymbol{p} - \alpha\boldsymbol{s_i})^T\boldsymbol{x} \\ & s.t. \qquad \boldsymbol{1}^T\boldsymbol{x} = 1; -\boldsymbol{x} \leq \boldsymbol{0},\end{aligned} \tag{4}$$

where $\boldsymbol{w}_i$ is the i-th column vector of the collaboration graph $\boldsymbol{W}$, $\boldsymbol{p} = [p_1, p_2, ..., p_K]^T \in \mathbb{R}^K$ is a relative dataset size vector, $\boldsymbol{s_i}$ is the i-th column vector of the similarity matrix $\boldsymbol{S}$. This quadratic program problem can be solved by conventional convex problem solver (Diamond & Boyd, 2016; Agrawal et al., 2018).

For higher computation efficiency in practice, cosine similarity of latter model parameters $\tilde{\boldsymbol{S}}_{ij} = \cos(\boldsymbol{\theta}_i[m:], \boldsymbol{\theta}_j[m:])$ can be used to approximate $\boldsymbol{S}_{ij}$, where $m \in [0, d]$ is an integer to balance the effectiveness and efficiency. In practice, m can be set to filter out the preceding layers and leave the last few fully-connected layers for similarity computation as we find that the last few fully-connected layers are sufficient to reflect the fine-grained model similarity under cases of data heterogeneity and model poisoning. This spirit accords with (Luo et al., 2021), which finds that the last few layers are more different in the case of data heterogeneity for conventional FL.

Then, the server can aggregate local models to obtain aggregated model for each client according to the optimized collaboration graph $\boldsymbol{W}$. For the ith client, the original aggregated model is $\widetilde{\boldsymbol{\theta}}_i = \sum_j \boldsymbol{W}_{ij}\boldsymbol{\theta}_j$ and the normalized aggregated model is $\bar{\boldsymbol{\theta}}_i = \sum_j \boldsymbol{W}_{ij}\boldsymbol{\theta}_j/||\boldsymbol{\theta}_j||$.

Optimizing local model at the client side. Based on the original aggregated model $\widetilde{\boldsymbol{\theta}}_i$ sent from the server, the ith client first initializes its local model: $\boldsymbol{\theta}_i \leftarrow \widetilde{\boldsymbol{\theta}}_i$, which explicitly injects collaboration knowledge to local model. A direct solver for (3) is to use the conventional SGD method, where $\bar{\boldsymbol{\theta}}_i$ is used to regularize the training of local model $\boldsymbol{\theta}_i$.

Algorithm 1 pFedGraph

```
Input: Total round $T$, client number $K$, learning rate $\eta$,
initial local models $\{\boldsymbol{\theta}_i^0\}_i$ and collaboration graph $\boldsymbol{W}^0$.
for each communication round $t = 1, ..., T$ do
    for each client $i = 1, ..., K$ in parallel do
        Server sends $\widetilde{\boldsymbol{\theta}}_i^t = \sum_j \boldsymbol{W}_{ij}^{t-1}\boldsymbol{\theta}_j^{t-1}$ to client $i$
        Initialize local model $\boldsymbol{\theta}_i^{t,0} \leftarrow \widetilde{\boldsymbol{\theta}}_i^t$
        Optimize local model at client side:
            Update $\boldsymbol{\theta}_i$ by minimizing $H_i(\boldsymbol{\theta}_i)$ defined in (3):
              $\boldsymbol{\theta}_i^{t,\tau} = \boldsymbol{\theta}_i^{t,\tau-1} - \eta\nabla H_i(\boldsymbol{\theta}_i^{t,\tau-1})$
    end for
    Optimize collaboration graph at server side:
        Update $\boldsymbol{W}^t$ by solving the quadratic program in (4)
end for
Return: Personalized models $\{\boldsymbol{\theta}_i\}_i$
```

After that, each client uploads its personalized model $\boldsymbol{\theta}_i$ to the server and start another FL round.

Note that this solution indicates that the server has to send both the original and normalized aggregated models ($\widetilde{\boldsymbol{\theta}}_i$ and $\bar{\boldsymbol{\theta}}_i$), which requires double downloading cost compared with vanilla FedAvg (McMahan et al., 2017) method. To reduce the downloading communication cost, we can slightly modify our pFedGraph algorithm: using the original aggregated model to replace the normalized aggregated model in the SGD updating step. That is, we use the original aggregated model $\widetilde{\boldsymbol{\theta}}_i$ to regularize local model training in solving (3). Then, the server only needs to send the aggregated model $\widetilde{\boldsymbol{\theta}}_i$ to client i, which requires the same downloading cost as vanilla FL method (McMahan et al., 2017). Experimental results validate that this modification does not affect the performances; see Table 7.

3.4. Discussions

Adaptation to different heterogeneity levels. Our proposed pFedGraph can automatically adjust the collaboration graph based on the measured model similarity, which is robust towards heterogeneity levels from two aspects. i) For extremely heterogeneous setting where neighbors' knowledge contributes little to each client's personalization, pFedGraph automatically assigns little collaboration weight for neighbors since the model similarity would be relatively small here. However, most methods (Li et al., 2021b; Collins et al., 2021) still collaborate with all clients, whose weights are based on dataset size. ii) In reverse, for extremely homogeneous setting where clients should deeply collaborate, pFedGraph automatically assigns relatively balanced collaboration weight for neighbors since the model similarity would be relatively high here. However, some methods (Arivazhagan et al., 2019; Collins et al., 2021) may still pursue too much personalization, leading to performance degradation compared to the already well-performed global model.

Robustness to model poisoning attacks. Our proposed pFedGraph is naturally robust towards model poisoning attacks, where malicious clients can upload arbitrary model parameters, since the model similarity can easily distinguish from a poisoned model and a benign model with a similar task such that pFedGraph will assign most weights to benign neighbors. However, most methods (Chen & Chao, 2021; Marfoq et al., 2022; T Dinh et al., 2020; Li et al., 2021b) based on the fully-collaborated global model are vulnerable to model poisoning since the global model has been polluted by the poisoning. Note that we do not discuss defense methods (Blanchard et al., 2017) here because they will affect the performance for scenarios without attacks, which preserves fair comparisons among original personalized FL methods.

Potential of handling different tasks. Our proposed pFedGraph focuses on model-level optimization, which makes no assumption on the targeting task, thus has the potential of handling different tasks such as classification (Hsu et al., 2020), regression (Mandal & Gong, 2019), detection (Liu et al., 2020), segmentation (Li et al., 2019), and recommendation (Harper & Konstan, 2015). However, some methods (Chen & Chao, 2021; Marfoq et al., 2022; Shamsian et al., 2021) are particularly designed for classification tasks.

4. Experiments

We list key implementation details and experimental results in this section and leave others to Appendix C.

4.1. Implementation Details

Datasets. Following most personalized FL literature (Li et al., 2021b; Collins et al., 2021; Marfoq et al., 2022), we conduct experiments on Fashion MNIST (Xiao et al., 2017), CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) for image classification tasks. Beyond this, we also use Yahoo! Answers text classification dataset (Zhang et al., 2015) to explore the performances of personalized FL methods in the realm of text modality.

Data heterogeneity. We consider several levels of data heterogeneity: i) pathological distribution (McMahan et al., 2017; Collins et al., 2021; Zhang et al., 2023), where each client is assigned with data from only 2 categories for CIFAR-10 and 20 categories for CIFAR-100. ii) Dirichlet distribution (Yurochkin et al., 2019; Wang et al., 2020a), where a distribution vector $\boldsymbol{q}_c \in \mathbb{R}^K$ is drawn from $Dir_K(\beta)$ for each category c and a $\boldsymbol{q}_{c,i}$ proportion of data samples of category c is assigned to client i. iii) Homogeneous distribution (McMahan et al., 2017), where each data sample is assigned to each client with equal probability

Table 1. Accuracy comparisons on classical image classification task under four different data heterogeneity levels (H-Level) illustrated in Section 4.1. pFedGraph consistently performs the best across different datasets on average.

Dataset	Fashion MNIST				Avg	CIFAR-10				Avg	CIFAR-100				Avg
H-Level	Extr.	Seve.	Mode.	Homo.		Extr.	Seve.	Mode.	Homo.		Extr.	Seve.	Mode.	Homo.	
Local	99.46	99.32	96.44	84.88	95.02	91.86	91.07	83.83	54.81	80.39	55.20	49.42	47.68	16.13	42.11
FedAvg	96.15	95.37	84.51	87.27	90.83	68.82	66.19	62.92	67.12	66.26	25.94	26.23	27.78	31.10	27.76
FedAvg-FT	99.25	99.09	96.33	86.23	95.23	90.02	90.20	84.09	63.09	81.85	51.35	51.12	50.58	25.47	44.63
FedProx	89.37	93.65	82.79	87.21	88.26	59.18	55.76	62.25	67.07	61.06	25.12	25.64	27.87	30.55	27.30
FedProx-FT	99.18	99.08	96.27	85.96	95.12	90.47	90.37	83.71	61.93	81.62	50.52	49.91	50.99	25.15	44.14
CFL	99.32	99.35	96.64	86.38	95.42	91.80	90.76	83.84	60.55	81.74	54.64	52.43	49.12	19.31	43.88
Per-FedAvg	99.18	99.15	96.30	86.28	95.23	90.35	89.60	84.02	63.24	81.80	52.14	51.14	50.38	25.18	44.71
pFedMe	98.40	98.34	93.34	77.32	91.85	82.15	81.73	75.24	47.48	71.65	34.10	33.48	34.37	13.18	28.78
FedAMP	98.51	98.71	93.13	74.42	91.19	80.45	86.90	75.49	45.49	72.08	36.68	37.50	31.04	10.07	28.82
Ditto	98.97	98.89	95.97	86.85	95.17	89.17	89.41	83.78	65.35	81.93	50.67	50.54	50.33	29.41	45.24
FedRep	99.24	99.17	96.44	85.73	95.15	91.01	90.02	83.47	62.88	81.85	51.21	51.72	50.15	21.53	43.65
pFedHN	98.95	98.88	95.87	80.11	93.45	90.51	89.91	82.57	62.78	81.44	49.87	49.06	49.08	25.94	43.49
FedRoD	99.26	99.16	96.46	85.50	95.10	91.03	90.66	83.49	62.07	81.81	51.30	49.91	47.96	18.71	41.97
kNN-Per	97.89	97.87	91.87	87.37	93.75	78.94	79.09	70.05	67.01	73.77	27.07	24.70	25.84	31.04	27.16
pFedGraph	99.46	99.39	96.46	87.25	**95.64**	92.61	92.74	84.28	67.37	**83.98**	54.84	56.79	51.63	31.16	**48.33**

$1/K$. Client number ranging from 5 to 50 is considered. Specifically, we denote the pathological distribution with $K = 5$ and 10 clients as extreme and severe, respectively; the Dirichlet distribution with $\beta = 0.1$ as modest.

Model Poisoning. Here, we consider model poisoning attacks, where malicious clients are intended to disturb the training process by sending arbitrary model parameters to the server. For a personalized FL system with K clients, suppose there are $r \cdot K$ malicious clients, where $r \in [0, 1]$ is the attack ratio. We consider 4 types of model poisoning attacks: parameters shuffling, same-value parameters, parameters sign flipping and Gaussian noise parameters (Lin et al., 2022; Li et al., 2021b).

Training setting. We consider 50 communication rounds in total as personalized FL is easier to converge, where each client runs for $\tau = 200$ iterations (Wang et al., 2020b). We use a simple CNN network with 3 convolutional layers and 3 fully-connected layers for image datasets (Li et al., 2021a). For text dataset, we use TextCNN (Zhang & Wallace, 2015; Zhu et al., 2020) model with 3 conv layers and a 256 dimension embedding layer. The optimizer used is SGD with learning rate 0.01 and a batch size 64.

Baselines. We compare 14 baselines, including local model training without collaboration (denoted by Local), FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020b) together with their fine-tuning (FT) versions, and 9 representative personalized FL methods. Among those personalized FL methods, CFL (Sattler et al., 2020) is based on clustering, Per-FedAvg (Fallah et al., 2020) is based on meta-learning, pFedMe (T Dinh et al., 2020) and Ditto (Li et al., 2021b) are based on regularized optimization, FedAMP (Huang et al., 2021) also adjusts aggregation manner, FedRep (Collins et al., 2021) is based on sharing backbone, FedRoD (Chen & Chao, 2021) is based on multi-branch architecture, and kNN-Per (Marfoq et al., 2022) is based on features memorization. We evaluate pFedGraph before and after training since pFedGraph re-initializes local models for each round. Hyper-parameters are tuned for each dataset and kept fixed for different levels of data heterogeneity; see the used hyper-parameters in Appendix C.1.4.

Table 2. Accuracy comparison on text classification task Yahoo! Answers under four different data heterogeneity levels (H-Level) illustrated in Section 4.1. pFedGraph consistently outperforms the others on text modality.

H-Level	Extr.	Seve.	Mode.	Homo.	Avg
Local	84.81	84.41	77.34	62.80	77.34
FedAvg	57.29	63.14	51.99	63.72	59.03
FedAvg-FT	83.48	87.02	80.24	63.87	78.65
FedProx	51.39	49.30	50.82	62.80	53.57
FedProx-FT	83.31	86.72	79.84	62.88	78.18
CFL	84.54	88.61	79.40	63.70	79.06
Per-FedAvg	71.93	86.92	80.16	63.66	75.66
pFedMe	61.44	64.16	58.91	22.69	51.80
FedAMP	65.50	76.85	64.26	22.59	57.30
Ditto	82.93	86.33	79.10	63.71	78.01
FedRep	82.79	86.00	76.29	46.42	72.87
FedRoD	82.56	86.51	78.82	62.35	77.56
pFedGraph	**85.03**	**89.04**	**80.56**	**63.88**	**79.62**

4.2. Personalized FL on Data Heterogeneity

Here, we compare our proposed pFedGraph with representative baselines under the context of data heterogeneity with different levels and two modalities (image and text). We also provide comparisons on recommendation task follow-

Table 3. Accuracy comparison on CIFAR-10 under different types of model poisoning attacks. The attack ratio r is 0.4. pFedGraph significantly outperforms the others.

Type	Shuffle	Same	Filp	Gauss	Avg
Local	54.98	54.86	54.88	54.90	54.91
FedAvg	42.55	12.77	9.92	11.08	19.08
FedAvg-FT	47.30	29.47	29.61	29.58	33.99
FedProx	42.47	11.97	9.87	10.57	18.72
FedProx-FT	46.09	29.34	28.87	29.19	33.37
CFL	62.13	29.47	60.90	29.69	45.55
Per-FedAvg	11.81	10.62	11.39	10.83	11.16
pFedMe	11.41	11.00	11.41	19.21	13.26
FedAMP	42.98	43.11	43.07	42.85	43.00
Ditto	44.53	10.07	12.39	10.71	19.43
FedRep	47.79	29.96	29.34	28.76	33.96
pFedHN	27.59	19.22	28.00	28.62	25.86
FedRoD	50.53	26.87	29.24	27.36	33.50
kNN-Per	47.20	11.73	28.98	12.33	25.06
pFedGraph	**65.45**	**66.36**	**66.25**	**65.46**	**65.88**

Table 4. Robustness towards different model poisoning attack ratios r. pFedGraph consistently outperforms the others and achieves significantly better performance on the most severe attack case.

Ratio	0.2	0.4	0.6	0.8
FedAvg	59.12	42.55	19.76	10.14
FedAvg-FT	55.79	47.30	33.41	30.65
Ditto	56.63	43.53	26.39	10.58
CFL	60.65	62.13	60.57	30.76
pFedGraph	**67.41**	**65.45**	**64.70**	**60.52**

ing pFL-Bench (Chen et al., 2022a) in Appendix C.2.

Results on image datasets. We conduct experiments under four heterogeneity levels on Fashion-MNIST, CIFAR-10 and CIFAR-100 dataset in Table 1. From the table, we see that i) pFedGraph consistently performs better or comparably across different heterogeneity levels and datasets while other methods only performs well for some cases, indicating that pFedGraph is robust towards different levels of heterogeneity. ii) On average, pFedGraph significantly outperforms others by 2% to 23%, indicating that pFedGraph can achieve competitive performance in personalized FL. Since we can not know the heterogeneity level in advance in practice, we believe that the averaged metric is a more critical evaluation protocol. We also conduct experiments by tuning β for Dirichlet distribution in Table 14.

Results on text dataset. Similarly, we conduct experiments under four different heterogeneity levels on text dataset Yahoo! Answers (Zhang et al., 2015) and report accuracy comparison in Table 2. From the table, we see that pFedGraph consistently performs best, indicating that pFedGraph is also adaptive towards different levels of heterogeneity on text modality.

Results on recommendation dataset. We follow pFL-Bench (Chen et al., 2022a) and conduct experiments on recommendation datasets (MovieLens1M/10M (Harper & Konstan, 2015))in Appendix C.2. Results show the applicability of pFedGraph on recommendation task.

4.3. Personalized FL on Model Poisoning Attack

Here, we compare our proposed pFedGraph with these baselines under the context of model poisoning attack. Four types of model poisoning are considered, where we set the constant as 1.0 for same-value poisoning and sample each parameter by $\mathcal{U} \sim (0, 1)$ for gauss-distribution poisoning. Note that we adopt hyper-parameters tuned in the previous experiments for all methods and the experiments are conducted on the homogeneous setting of CIFAR-10.

We conduct performance comparisons from two aspects: different poisoning types and different attack ratios.

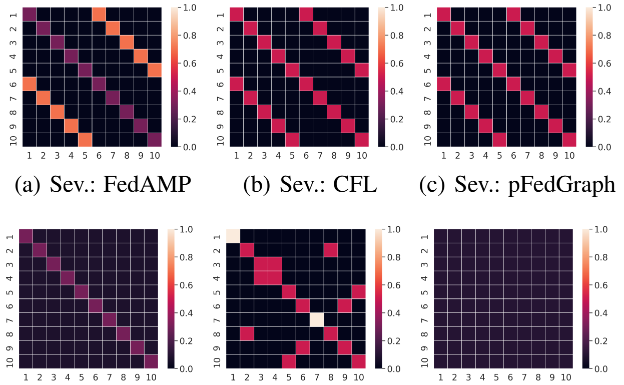
Robustness towards different types of model poisoning. We report the performances achieved by each method under four types of model poisoning attack in Table 3. From the experiments, we see that i) On average, all personalized FL baselines perform worse than local training, which indicates that the attack can affects could be a key issue in personalized FL and that a robust personalized FL method is needed. ii) pFedGraph consistently performs the best across different types of poisoning attacks and significantly outperforms all baselines on average. Note that we do not compare with defense methods (Blanchard et al., 2017) as they could limit the performance under the scenario without attack while we can not know whether attack exists in advance.

Robustness towards different model poisoning attack ratios. Here in Table 4, we compare pFedGraph with four representative methods, FedAvg, FedAvg-FT, Ditto and CFL under four different attack ratios ranging from 0.2 to 0.8. From the table, we see that i) pFedGraph consistently and significantly outperforms others and ii) pFedGraph achieves largest performance gain under the most severe case, indicating that pFedGraph is robust towards different model poisoning attack ratios.

4.4. Visualization of Collaboration Graph

Here, we visualize the collaboration graph of several representative methods. On CIFAR-10, We consider i) severe and homogeneous settings; ii) sign-flipping model poisoning.

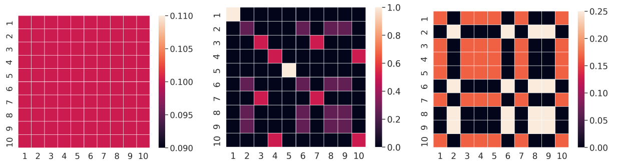
Figure 2 shows that i) under the severe heterogeneous setting, where there are 5 pairs of clients that share similar data distribution, CFL and pFedGraph both well capture the collaboration relationships and assign equal collaboration weight. ii) However, for homogeneous setting, where all



(a) Sev.: FedAMP (b) Sev.: CFL (c) Sev.: pFedGraph

(d) Hom.: FedAMP (e) Hom.: CFL (f) Hom.:pFedGraph

Figure 2. Visualization of collaboration graph under severe (Sev.) and homogeneous (Hom.) settings. pFedGraph is adaptive to different data heterogeneity levels.



(a) Vanilla (b) CFL (c) pFedGraph

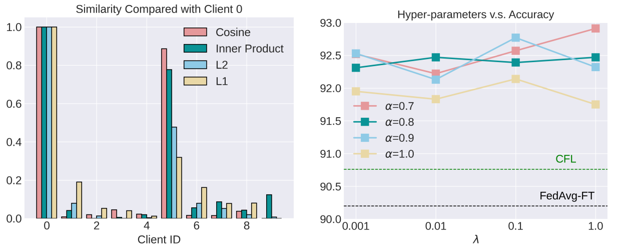
Figure 3. Visualization of collaboration graph under sign-flipping attack, where client $2, 6, 8, 9$ are malicious and the other clients share similar data distribution. pFedGraph successfully construct full-collaboration within the benign clients.

clients should fully collaborate, CFL fails to assign equal weights to all clients. Figure 3 shows that under an attack scenario, where client $2, 6, 8, 9$ are malicious and the others share similar distributions. pFedGraph successfully constructs a full collaboration graph among the benign clients and filters out malicious clients.

4.5. Effects of Model Similarity Metric

One key design in pFedGraph is that we apply cosine similarity to optimize the collaboration graph and regularize local model training. Here, we compare four types of model similarity metrics, including cosine-based, inner-product-based, ℓ_2-based and ℓ_1-based metrics.

We conduct experiments on the severe case on CIFAR-10 where client 0 and 5 share similar data distributions. Then, we measure the similarity between local model of Client 0 and all clients. Note that we normalize the similarity values based on the value of self-similarity of Client 0 for more clear comparisons. We plot the results in Figure 4(a). From the figure, we see that cosine similarity best captures the similarity between client 0 and 5, and produces the most distinguishable similarity values among all clients.



(a) Different Metrics (b) Different Hyper-parameters

Figure 4. Effects of model similarity metrics and hyper-parameters. (a) Cosine similarity best captures the relationships between client 0 and 5 since they share similar data distributions. (b) pFedGraph is insensitive to hyper-parameters.

Table 5. Effects of model difference regularization. InnerP. denotes inner product. Our proposed cosine-based regularization consistently performs the best on average.

H-Level	Extreme	Severe	Modest	Homo.	Avg
Cosine	92.55	92.74	84.11	67.37	**84.19**
InnerP.	92.28	92.76	74.81	67.58	81.86
ℓ_2	91.80	91.49	74.96	64.83	80.77
ℓ_1	91.93	90.76	74.63	64.72	80.51

Additionally, we conduct experiments on all cases on CIFAR-10 and compare the achieved accuracy of these measuring metrics. Results are reported in Table 5. From the table, we see that the proposed cosine-based optimization tends to perform the best.

4.6. Ablation Study

Personalized FL at different scale. On a scenario similar to the severe case on CIFAR-10, we adjust the number of clients in FL system to show the comparison of personalized FL methods at different scale. From Table 6, we see that pFedGraph consistently performs the best across different client numbers, indicating the effectiveness of pFedGraph at different scale.

Effects of approximation. Here, we explore the effects of approximation on the performance of our proposed pFedGraph. There are two approximations: 1) replacing the normalized aggregated model in Equation (3) with unnormalized aggregated model for the sake of communication efficiency; and 2) cosine similarity measurement based on partial model parameters for the sake of computation efficiency. Experiments are conducted on CIFAR-10 and results are reported in Table 7. Aggreg. and Measure. denote the first and second approximations, respectively. From the table, we see that these two approximations can improve the efficiency while hardly affects performance across different heterogeneous settings.

Table 6. Comparisons at various client numbers. pFedGraph consistently performs the best.

Number	10	20	30	50
Local	91.07	89.21	88.07	86.18
FedAvg	66.19	62.98	65.73	63.13
FedAvg-FT	90.20	90.13	89.62	89.39
FedProx	55.76	58.45	62.92	62.75
FedProx-FT	90.37	89.94	89.87	89.15
CFL	90.76	91.63	91.77	90.42
Per-FedAvg	89.60	88.88	89.84	89.18
pFedMe	81.73	81.34	81.06	80.56
FedAMP	86.90	86.72	86.42	86.26
Ditto	89.41	89.31	89.25	88.95
FedRep	90.02	90.24	89.80	88.91
pFedHN	89.91	88.98	88.48	87.72
FedRoD	90.66	90.37	89.93	88.94
KNN-Per	79.09	78.48	77.42	76.47
pFedGraph	**92.35**	**92.59**	**92.10**	**91.77**

Table 7. Effects of approximation. 1) Replacing normalized aggregated model with unnormalized aggregated model for regularization; 2) Cosine similarity measured on partial model parameters. Approximation basically does not affect the performance.

Aggreg.	Measure.	Extreme	Severe	Modest	Homo.
×	×	92.14	92.31	84.11	67.10
×	√	92.12	92.74	84.28	67.37
√	×	92.55	92.54	84.26	66.55
√	√	92.61	92.35	84.21	66.70

Effects of optimizing based on both cosine similarity and dataset size. In Equation (2), we optimize aggregation weights based on both relative dataset size and the similarities between clients and thus the learned collaboration graph can capture both the information of dataset size and model similarity. To verify the effectiveness of such design, we conduct the experiments on CIFAR-100 by directly using cosine similarities as the collaboration weight for comparisons in Table 8. From the table, we see that our proposed optimization in Equation (2) contributes to significantly better performance than directly using cosine similarities. See more results in Table 13.

Effects of using dataset size for optimization. We apply relative dataset size in the first term of Equation (2). Here, we explore its effectiveness by comparing with replacing p_k with $1/K$, where K is client number. We conduct experiments Dirichlet-distribution-based case on CIFAR-10 and tune β for more comprehensive observations. The results are reported in Table 9. From the table, we see that using relative dataset size for optimizing collaboration graph consistently performs better than simply using $1/K$.

Table 8. Optimization based on both similarity and dataset size in Equation (2) brings significant performance gain compared with directly using similarity as aggregation weight. Experiments are conducted on CIFAR-100, see more in Table 13.

H-Level	Extreme	Severe	Modest	Homo.
Only Similarity	38.26	50.23	50.83	30.82
Similarity & Size	**54.35**	**56.79**	**51.29**	**30.89**

Table 9. Using relative dataset size p_k for optimizing collaboration graph contributes to better performance than simply using equal weight $1/K$. Experiments are conducted under different levels of heterogeneity (controlled by β).

β	0.1	0.2	0.3	0.4	0.5
$1/K$	83.43	83.61	76.99	75.74	75.05
$\boldsymbol{p_k}$	**84.26**	**83.92**	**77.26**	**76.65**	**75.42**

Effects of hyper-parameters. We tune hyper-parameters $\alpha \in \{0.7, 0.8, 0.9, 1.0\}$ for optimizing collaboration graph and $\lambda \in \{0.001, 0.01, 0.1, 1.0\}$ for regularizing local model training under the severe case on CIFAR-10. We show the results in Figure 4(b) together with results of FedAvg-FT and CFL as reference. From the figure, we see that i) for a wide range of hyper-parameters, our proposed pFedGraph consistently outperforms the baselines; ii) our proposed pFedGraph is not sensitive to hyper-parameters, while $\alpha = 0.7 \sim 0.9$ and $\lambda = 0.1 \sim 1.0$ tends to perform better.

5. Conclusions

A fundamental challenge in personalized FL is an appropriate tradeoff between the individual utilities and the benefits through collaboration. In this work, we handle this issue by proposing a personalized FL method pFedGraph, which consists of two key modules: i) inferring collaboration graph at the server side that promotes fine-grained collaboration and ii) optimizing local model with the assistance of aggregated model at the client side that promotes personalization. The proposed pFedGraph has strong ability to adapt to diverse data heterogeneity levels, as the proposed collaboration graph always pushes each client to collaborate more with similar and beneficial clients, promoting local data homogeneity. Extensive experiments show that pFedGraph can adaptively work well under different data heterogeneity levels and cases with malicious clients.

Acknowledgements

This research is supported by the National Key R&D Program of China under Grant 2021ZD0112801, NSFC under Grant 62171276 and the Science and Technology Commission of Shanghai Municipal under Grant 21511100900 and 22DZ2229005. Thanks feedback from Jianyu Wang.

References

Acar, D. A. E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., and Saligrama, V. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2020.

Agrawal, A., Verschueren, R., Diamond, S., and Boyd, S. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Arivazhagan, M. G., Aggarwal, V., Singh, A. K., and Choudhary, S. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Baek, J., Jeong, W., Jin, J., Yoon, J., and Hwang, S. J. Personalized subgraph federated learning. *arXiv preprint arXiv:2206.10206*, 2022.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017.

Chen, D., Gao, D., Kuang, W., Li, Y., and Ding, B. pfl-bench: A comprehensive benchmark for personalized federated learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022a.

Chen, F., Long, G., Wu, Z., Zhou, T., and Jiang, J. Personalized federated learning with graph. *arXiv preprint arXiv:2203.00829*, 2022b.

Chen, H.-Y. and Chao, W.-L. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2021.

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

Deng, Y., Kamani, M. M., and Mahdavi, M. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

Diamond, S. and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

Hanzely, F. and Richtárik, P. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Hsu, T.-M. H., Qi, H., and Brown, M. Federated visual classification with real-world data distribution. In *European Conference on Computer Vision*, pp. 76–92. Springer, 2020.

Huang, Y., Chu, L., Zhou, Z., Wang, L., Liu, J., Pei, J., and Zhang, Y. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7865–7873, 2021.

Jeong, W. and Hwang, S. J. Factorized-fl: Personalized federated learning with parameter factorization & similarity matching. In *Advances in Neural Information Processing Systems*, 2022.

Jiang, Y., Konečnỳ, J., Rush, K., and Kannan, S. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

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

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

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

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020a.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020b.

Li, T., Hu, S., Beirami, A., and Smith, V. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021b.

Li, W., Milletarì, F., Xu, D., Rieke, N., Hancox, J., Zhu, W., Baust, M., Cheng, Y., Ourselin, S., Cardoso, M. J., et al. Privacy-preserving federated brain tumour segmentation. In *International workshop on machine learning in medical imaging*, pp. 133–141. Springer, 2019.

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

Lin, S., Han, Y., Li, X., and Zhang, Z. Personalized federated learning towards communication efficiency, robustness and fairness. *Advances in Neural Information Processing Systems*, 2022.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33: 2351–2363, 2020.

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

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

Mandal, K. and Gong, G. Privfl: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pp. 57–68, 2019.

Marfoq, O., Neglia, G., Vidal, R., and Kameni, L. Personalized federated learning through local memorization. In *International Conference on Machine Learning*, pp. 15070–15092. PMLR, 2022.

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

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

Sattler, F., Müller, K.-R., and Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

Shamsian, A., Navon, A., Fetaya, E., and Chechik, G. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pp. 9489–9502. PMLR, 2021.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017.

T Dinh, C., Tran, N., and Nguyen, J. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaeni, Y. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=BkluqlSFDS.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020b.

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Yang, T., Andrew, G., Eichner, H., Sun, H., Li, W., Kong, N., Ramage, D., and Beaufays, F. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

Ye, R., Ni, Z., Xu, C., Wang, J., Chen, S., and Eldar, Y. C. Fedfm: Anchor-based feature matching for data heterogeneity in federated learning. *arXiv preprint arXiv:2210.07615*, 2022.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., and Khazaeni, Y. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pp. 7252–7261. PMLR, 2019.

Zhang, J., Guo, S., Ma, X., Wang, H., Xu, W., and Wu, F. Parameterized knowledge transfer for personalized federated learning. *Advances in Neural Information Processing Systems*, 34:10092–10104, 2021.

Zhang, J., Hua, Y., Wang, H., Song, T., Xue, Z., Ma, R., and Guan, H. FedALA: Adaptive Local Aggregation for Personalized Federated Learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zhang, Y. and Wallace, B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

Zhu, X., Wang, J., Hong, Z., and Xiao, J. Empirical studies of institutional federated learning for natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 625–634, 2020.

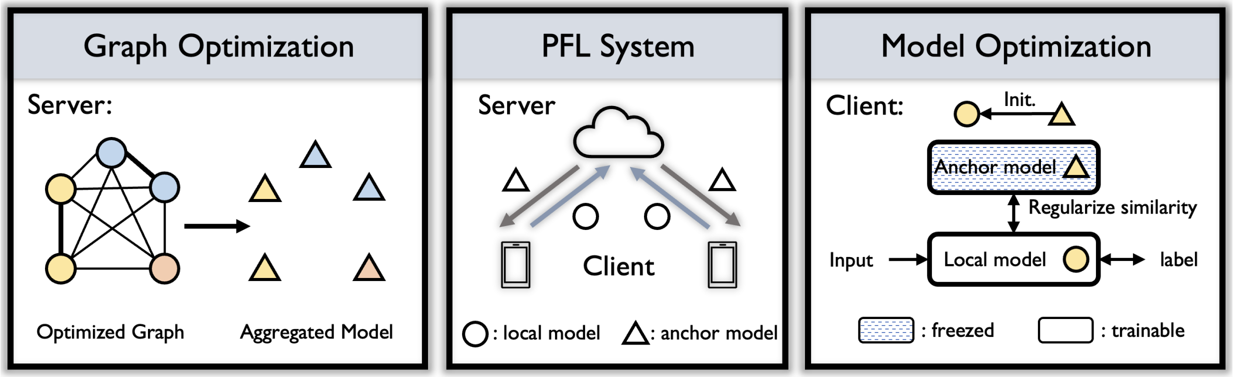


Figure 5. The overview of pFedGraph algorithm. pFedGraph consists of two key procedures: collaboration graph optimization for model aggregating at the server side and local model optimization regularized by the aggregated model at the client side.

A. Methodology

We provide a overview of the pFedGraph algorithm in Figure 5.

A.1. Explanations for Obtaining Equation (2)

We provide the rationales of approximation from Equation (1) to Equation (2).

Equation (1) is an ideal optimization problem, which can not be directly solved due to practical restrictions. To meet the practical privacy restrictions, problem approximations (or transformations) are required, which unavoidably leads to difficulties in proving equivalence between the original and approximated form. Here, we can explain the rationale behind our approximation (or transformation) theoretically and empirically.

From the perspective of theory. The goal of FedAvg (McMahan et al., 2017) is to obtain a global model that fits well to all clients data, that is, relatively low value evaluated on each client's local loss function. We firstly rewrite the optimization problem of FedAvg in the form of first term in our Equation (1):

$$\min_{\{\boldsymbol{\theta}_i\},\boldsymbol{W}} \sum_{i=1}^{K} p_i F_i(\sum_{j=1}^{K} \boldsymbol{W}_{ij}\boldsymbol{\theta}_j) \quad s.t. \;\; \boldsymbol{W}_{1j} = \boldsymbol{W}_{2j} = ... = \boldsymbol{W}_{Kj}, \quad \sum_{j=1}^{K} \boldsymbol{W}_{ij} = 1, \forall i. \tag{5}$$

In FedAvg (McMahan et al., 2017), considering the variable of aggregation weight $\boldsymbol{W}$, the optimal solution is $\boldsymbol{W}_{1j} = \boldsymbol{W}_{2j} = ... = \boldsymbol{W}_{Kj} = p_j$ (otherwise the problem of inconsistent objective will raise (Wang et al., 2020b)), which is directly applied by most FL methods (McMahan et al., 2017; Li et al., 2021b). Considering the first term in our approximated Equation (2) the optimal solution is also $\boldsymbol{W}_{1j} = \boldsymbol{W}_{2j} = ... = \boldsymbol{W}_{Kj} = p_j$, which aligns with the solution in FedAvg. This is reasonable since model aggregated by these set of aggregation weights is a safe choice for all clients similar to FedAvg.

From the perspective of experiments. From the experiments, we know that FedAvg-FT (FedAvg with fine-tuning) is a strong baseline, indicating that our first term in approximated Equation (2) is reasonable as it also tends to assign larger weight to clients with more data samples as FedAvg-FT. Also, for example, when the data distributions are homogeneous, we know that the optimal strategy is assigning aggregation weights according to dataset size, and experiments show that our method produces exactly the same aggregation weights as this optimal strategy.

Our focus in this paper is to propose a practical personalized FL method guided by Equation (1) and our experiments have verified its effectiveness. Considering provable approximation, one approximated approach that has potential to prove the equivalence is to introduce a public dataset such that the value of local loss function can be approximated at the server side. We did not consider such approach for the sake of practicability as the public dataset could be not existed in practice. More theoretical analysis could be our future work.

B. Detailed Algorithm Comparisons

B.1. Comparison with FedAMP (Huang et al., 2021).

Though intuitively, we are trying to achieve a similar goal in personalized FL, our paper is different from the formulation of optimization problems fundamentally, followed by algorithm design and algorithm performance across different settings.

Optimization problem. The optimization problem in FedAMP (Huang et al., 2021) does not consider model re-initialization at the client side after receiving an aggregated model from the server, where the optimizing variables are local models only. The optimization problem in ours considers model re-initialization at the client side, where the optimizing variables are local models and collaboration graph. In detail, the local loss function is evaluated at variable of corresponding local model in optimization problem in FedAMP (Huang et al., 2021), while the local loss function is evaluated at aggregated model in our optimization problem. The reason for such design is that model re-initialization is critical in many cases, especially when clients share similar data distributions, as the model parameters contain a lot of informative knowledge. Also, our method explicitly optimizes the collaboration graph, which extends its interpretability as we can explicitly see their collaboration relationships.

Algorithm design. 1) FedAMP (Huang et al., 2021) applies a single gradient descent step to obtain the aggregated model, which relies on manually chosen small step size. While in our proposed constrained optimization problem, the collaboration graphs can be easily solved by conventional convex problem solver. 2) Optimization in FedAMP is entirely based on model difference shile neglecting the effects of dataset size. However, the optimization in our method takes both model difference and dataset size into account, which is more reasonable as neighbors with larger datasets and smaller differences deserve higher collaboration intensity. For example, when two neighbors have the same similarity levels with a client, the neighbor with larger dataset size should deserve a higher collaboration intensity. 3) FedAMP does not re-initialize the local model with the received aggregated model which fails to fully utilize the beneficial knowledge in the aggregated model, while we will re-initialize the local model before launching local model training. Such difference could bring significant impact, as in some cases where multiple clients share similar data distributions and thus all local models should share the same global model. In such cases, all clients should re-initialize their local models with the same global model.

Performance. Under different levels of data heterogeneity, we consistently see that our method performs better. The reason can be that 1) the aggregation weights in FedAMP are not accurate enough to capture optimal collaboration relationships as it only takes one small step of optimizing; 2) FedAMP does not reinitilize the local model after receiving the aggregated model.

Table 10. Comparisons with FedAMP and HeurFedAMP.

SETTING	CIFAR-10-SEVE.	CIFAR-10-HOMO.	CIFAR-100-SEVE.	CIFAR-100-HOMO.
FEDAMP	85.52	42.91	35.61	8.42
HEURFEDAMP	86.90	45.49	37.50	10.07
PFEDGRAPH	**92.54**	**66.55**	**56.79**	**30.89**

B.2. Comparison with CFL (Sattler et al., 2020)

CFL (Sattler et al., 2020) is specifically designed to create isolated clusters where clients within the same cluster have uniform edge weights assigned to them. In CFL, collaboration weights between clients belonging to different clusters are assigned zero, thus resulting in no collaboration between them. In contrast, our proposed pFedgraph approach utilizes a global collaboration graph that assigns edge weights based on both the dataset sizes and the similarities between clients, resulting in a much more fine-grained collaboration graph and enhancing the flexibility.

B.3. Comparison with SFL (Chen et al., 2022b)

SFL (Chen et al., 2022b) primarily focuses on the use of graph convolution networks to address situations where a collaboration graph has already been established, such as the road device topology that is pre-established in a smart city. However, our method aims to infer the collaboration graph between clients in order to tackle the personalized federated learning setting, where a prior graph is not readily available.

Although this paper have attempted to adapt their approach to the missing prior graph scenario, their proposed adaptation

Table 11. Comparisons with SFL (Chen et al., 2022b).

DATASET	CIFAR-10				CIFAR-100				YAHOO! ANSWERS			
H-LEVEL	EXTR.	SEVE.	MODE.	HOMO.	EXTR.	SEVE.	MODE.	HOMO.	EXTR.	SEVE.	MODE.	HOMO.
SFL*	78.50	89.43	83.26	66.36	38.26	50.23	50.83	30.82	76.88	86.75	79.64	59.99
PFEDGRAPH	**92.55**	**92.54**	**84.26**	**66.55**	**54.35**	**56.79**	**51.29**	**30.89**	**85.03**	**89.04**	**80.56**	**63.88**

Table 12. Comparison on recommendation datasets MovieLens1M and MovieLens10M. Test loss is reported. Lower is better. Results show that pFedGraph also performs well for recommendation task.

METHOD	LOCAL	FEDAVG	FEDAVG-FT	FEDPROX	FEDPROX-FT	DITTO	PERFEDAVG	**PFEDGRAPH**
MOVIELENS1M	0.8014	0.7920	0.7910	0.7962	0.7951	0.8009	0.7904	**0.7896**
MOVIELENS10M	0.9508	0.9534	0.9494	0.9626	0.9585	0.9502	0.9485	**0.9483**

suffers from computational inefficiency, as the use of a graph convolution network becomes meaningless in the absence of a prior graph. Furthermore, the performance of this adaptation is subpar, as evidenced by the results of SFL* in (Chen et al., 2022b). To provide a more comprehensive comparison, we conducted experiments and found that our pFedgraph consistently outperforms SFL* as shown in Table 11.

B.4. Comparison with Factorized-FL (Jeong & Hwang, 2022) and FED-PUB (Baek et al., 2022)

Though these two works are both relevant to the broader field of federated learning and graph, they are focused on distinctly different topics from our method.

Factorized-FL (Jeong & Hwang, 2022) is specifically designed for the case where clients have their own personalized labels incompatible with those from other clients, which is significantly different from ours as we consider standard personalized task where all clients share the same set of labels. FED-PUB (Baek et al., 2022) is specifically designed for subgraph federated learning and Graph Neural Network where the 'graph' in this paper is related to task, while the 'graph' in ours is used to model collaboration among clients.

They measure similarity but directly use similarity as the aggregation weight, while we propose a novel optimization problem for learning the weight in the collaboration graph, which considers both the similarity and the relative dataset size. Considering both similarity and dataset size is critical as models trained by using more data samples tend to carry much more informative information.

C. Experiments

C.1. Implementation Details

C.1.1. MODEL

For Fashion MNIST (Xiao et al., 2017), CIFAR-10(Krizhevsky et al., 2009) and CIFAR-100(Krizhevsky et al., 2009), we use a simple CNN network with 3 convolutional layers and 3 fully-connected layers for image datasets (Li et al., 2021a). For Yahoo! News dataset, we use a TextCNN (Zhang & Wallace, 2015; Zhu et al., 2020) model with 3 conv layers. There is 1 channel and 100 kernels each layer, with kernal size $[3, 4, 5]$. The dimension of embedding layer is 256 and the weight of embedding layer is initialized randomly from a uniform distribution $[-0.1, 0.1]$.

C.1.2. TRAINING

We run 50 communication rounds for all experiments. In each round, every client runs for $\tau = 200$ iterations (Wang et al., 2020b) with a batch size of 64. We use SGD optimizer with learning rate 0.01, weight decay rate $1e^{-5}$ and SGD momentum 0.9. These are commonly used experimental settings (Li et al., 2021a; Luo et al., 2021). For each client, 20% of the training set is held out for validation. We average the results on each local validation set and save the best model. Finally, we report the testing accuracy of the best model on the testing dataset.

Table 13. Optimization based on both similarity and dataset size in Equation (2) brings significant performance gain compared with directly using similarity as aggregation weight.

Dataset	CIFAR-10				CIFAR-100				Yahoo! Answers			
H-Level	Extr.	Seve.	Mode.	Homo.	Extr.	Seve.	Mode.	Homo.	Extr.	Seve.	Mode.	Homo.
Only Similarity	90.53	90.03	83.26	66.36	38.26	50.23	50.83	30.82	76.88	86.75	79.64	59.99
Similarity & Size	**92.55**	**92.54**	**84.26**	**66.55**	**54.35**	**56.79**	**51.29**	**30.89**	**85.03**	**89.04**	**80.56**	**63.88**

Table 14. Accuracy comparisons under different parameter β of Dirichlet distribution.

Dataset	CIFAR-10				CIFAR-100			
β	0.01	0.1	0.5	5.0	0.01	0.1	0.5	5.0
Local	**97.86**	83.83	72.08	57.15	**64.34**	47.68	30.56	18.61
FedAvg	71.50	62.92	63.00	64.85	24.08	27.78	30.18	30.76
FedAvg-FT	96.92	84.09	75.22	64.20	63.80	50.58	37.44	27.17
FedProx	64.72	62.25	22.08	64.70	23.76	27.87	30.02	31.00
FedProx-FT	96.79	83.71	74.85	63.79	62.72	50.99	37.03	27.41
CFL	97.65	83.84	74.91	62.51	63.98	49.12	32.21	20.32
PerFedAvg	96.97	84.02	74.40	63.82	63.15	50.38	35.62	26.68
pFedMe	95.10	75.24	66.26	49.86	46.19	34.37	23.09	14.85
FedAMP	93.68	75.49	64.96	49.08	42.79	31.04	20.52	12.01
Ditto	96.74	83.78	75.97	65.95	61.46	50.33	37.02	30.37
FedRep	97.33	83.47	74.79	64.59	63.82	50.15	35.10	23.61
pFedHN	97.09	82.57	74.79	64.40	62.51	49.08	36.30	26.84
FedRod	97.07	83.49	73.80	63.24	62.83	47.96	32.24	20.15
KNN-Per	95.11	70.05	62.41	65.82	38.02	25.84	24.14	31.12
pFedGraph	97.66	**84.26**	**75.42**	**66.43**	64.23	**51.29**	**37.52**	**31.22**

C.1.3. Datasets

Following most personalized FL literature (Li et al., 2021b; Collins et al., 2021; Marfoq et al., 2022), we conduct experiments on Fashion MNIST (Xiao et al., 2017), CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) for image classification tasks. We also use Yahoo! Answers text classification dataset (Zhang et al., 2015) to explore the performances of personalized FL methods in the realm of text modality. To accelerate the training process, we sample 3072 training samples for each client.

C.1.4. Details of FL methods

For pFedGraph, we set $\lambda = 0.01$. For α, since the absolute magnitude will be affected by client number K, α should be set proportional to K. Generally, $\alpha = K * 0.08$ is a pleasant choice. We empirically calculate the similarity based on current model subtracted by initial global model (Huang et al., 2021; Sattler et al., 2020). We also empirically set the similarity values that is larger than 0.9 as 1.0. The rationale is that even if two models trained under the same distribution are different, thus we regard two models with such high similarity (i.e., 0.9) as fully shareable (i.e., 1.0). For FedProx (Li et al., 2020b) and FedProx-FT, we set $\lambda = 0.01$. For CFL (Sattler et al., 2020), we set $\epsilon_1 = 2.0$ and $\epsilon_2 = 2.5$. For Per-FedAvg (Fallah et al., 2020), we set $\alpha = 0.001$ and $\beta = 0.01$. For pFedMe (T Dinh et al., 2020) , we set $\eta = 0.01$, $\beta = 1.0$, $\lambda = 15$ and $R = 5$. For FedAMP (Huang et al., 2021) , we use the heuristic improvement version and set $\lambda = 1$, $\sigma = 10$, $\varepsilon_{i,i} = 0.3$. For Ditto(Li et al., 2021b), we use $\lambda = 1$. For pFedHN (Shamsian et al., 2021), we follow the setting as original paper. For KNN-Per (Marfoq et al., 2022), we set use the default hyper-parameter with $\lambda = 0.5$.

C.2. Personalized FL on Recommendation.

Here, we extend comparisons to recommendation task for more comprehensive comparisons. For this experiment, we follow the personalized FL benchmark pFL-Bench (Chen et al., 2022a) and utilize two datasets: MovieLens1M and MovieLens10M. Results in Table 12 show that pFedGraph is also applicable to recommendation task.

C.3. Effects of optimizing based on both cosine similarity and dataset size

Here, we provide the full table evaluated on CIFAR-10, CIFAR-100 and YAHOO! Answers in Table 13, as a complement of Table 8.

C.4. Effects of Different Parameter of Dirichlet Distribution

Here, we tune the parameter of Dirichlet distribution β. Note that a smaller β denotes a more heterogeneous scenario. Experiments are conducted on CIFAR-10 and CIFAR-100. Results are reported in Table 14. We see that 1) under the most heterogeneous setting (i.e., $\beta = 0.01$), Local performs the best while ours perform on par with Local. This is reasonable as when the heterogeneity is high, clients may have little information to share and to benefit others, thus Local tends to perform well. Though pFedGraph successfully learns to isolate each client, it has a regularization term during training, which could affect the training speed and thus pFedGraph performs slightly worse than Local. However, our pFedGraph performs better than other personalized FL methods. 2) Under other settings, pFedGraph consistently performs the best.