PGFed: Personalize Each Client's Global Objective for Federated Learning

Jun Luo[1] Matias Mendieta[2] Chen Chen[2] Shandong Wu[1,3,4,5]
[1] Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA
[2] Center for Research in Computer Vision, University of Central Florida, Orlando, FL
[3] Department of Radiology, University of Pittsburgh, Pittsburgh, PA
[4] Department of Biomedical Informatics, University of Pittsburgh, Pittsburgh, PA
[5] Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA
jul117@pitt.edu ma584394@ucf.edu chen.chen@crcv.ucf.edu wus3@upmc.edu

Abstract

Personalized federated learning has received an upsurge of attention due to the mediocre performance of conventional federated learning (FL) over heterogeneous data. Unlike conventional FL which trains a single global consensus model, personalized FL allows different models for different clients. However, existing personalized FL algorithms only ***implicitly*** *transfer the collaborative knowledge across the federation by embedding the knowledge into the aggregated model or regularization. We observed that this implicit knowledge transfer fails to maximize the potential of each client's empirical risk toward other clients. Based on our observation, in this work, we propose* ***P****ersonalized* ***G****lobal* ***Fed****erated Learning (`PGFed`), a novel personalized FL framework that enables each client to* ***personalize*** *its own* ***global*** *objective by* ***explicitly*** *and adaptively aggregating the empirical risks of itself and other clients. To avoid massive ($O(N^2)$) communication overhead and potential privacy leakage while achieving this, each client's risk is estimated through a first-order approximation for other clients' adaptive risk aggregation. On top of `PGFed`, we develop a momentum upgrade, dubbed `PGFedMo`, to more efficiently utilize clients' empirical risks. Our extensive experiments on four datasets under different federated settings show consistent improvements of `PGFed` over previous state-of-the-art methods. The code is publicly available at* `https://github.com/ljaiverson/pgfed`.

1. Introduction

Recent years have witnessed the prosperity of federated learning (FL) [26, 14, 21, 39] in collaborative machine learning where the participating clients are subject to strict privacy rules [38]. Conventional FL aims to train a single global consensus model by orchestrating the participating clients with a central server. The most notable FL algorithm, `FedAvg` [26], proceeds the training by exchanging between clients and server only the locally updated and globally aggregated model weights, leaving the private datasets intact. As a privacy-preserving machine learning technique, FL tremendously boosts new collaborations of decentralized parties in a number of areas [41, 16, 24, 14].

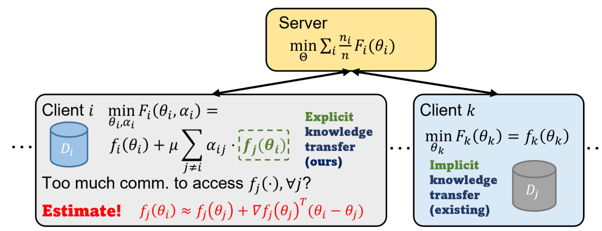


Figure 1. In `PGFed`, the explicit collaborative knowledge transfer shows in the design of the local objective as a "personalized global" objective at client i, where non-local risks are involved.

Unfortunately, along with the emergence of conventional FL, new challenges have been posed in terms of systems and statistical heterogeneity [21]. Systems heterogeneity addresses the variability of clients' computation abilities, sizes of storage, or even the choices of model architecture due to different hardware constraints of the clients. Whereas statistical heterogeneity, **the focus of this work**, refers to the non-IID data among the clients, which could lead to non-guaranteed convergence [22, 30] and poor generalizability performance [32], even after fine-tuning [13].

Over the years, the mediocre performance of conventional FL over heterogeneous data has not only promoted solutions that improve the single consensus model on top of `FedAvg` [22, 1, 15, 20], a new paradigm, ***personalized FL*** [36, 18], has emerged as well. In this paradigm, personalized models are allowed for each individual client. Some efforts in this direction focus on different personalized layers and optimization techniques [23, 5, 29]. Leveraging multi-task [44] or meta-learning [10] is also shown to be beneficial in personalized FL [34, 7, 3]. Other works include clustered FL [9, 33, 31], interpolation of personalized models [43, 25, 6], and fine-tuning [28, 19, 40].

However, in most existing personalized FL algorithms [23, 5, 6, 12, 2, 7, 4, 28], *the way in which the* ***collaborative knowledge*** *is transferred from the server to the clients is* ***implicit***. Here, we consider the collaborative knowledge as non-local information, such as the global objective of FedAvg, $F(\boldsymbol{\theta}) = \sum_i p_i F_i(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the global model and $F_i(\cdot)$ represents client i's local objective whose weights are denoted as p_i's. In addition, we define *"implicitness"* by defining its opposite side, *"explicitness"*, as a direct engagement with multiple clients' empirical risks. For instance, updating the global model of FedAvg is explicit, where the direct engagement is achieved through communication. However, this can hardly be the case for updating clients' personalized models, as it would take $O(N^2)$ communication overhead to transmit each client's personalized model to every client, assuming FedAvg's communication cost is $O(N)$ over N clients. Consequently, most personalized FL algorithms implicitly transfer the collaborative knowledge from the server to the clients by embedding it into the aggregation of model weights or as different kinds of regularizers.

Why should we care about the "*explicitness*", especially for updating the personalized models? Let us first assume all communication has zero cost, and, as an example, design a client's explicit local objective as a *"personalized global objective"* in the same form as the global objective of FedAvg (weighted sum of all clients' risks), i.e. $F_i(\boldsymbol{\theta}_i) = f_i(\boldsymbol{\theta}_i) + \mu/(N-1)\sum_{j\neq i} f_j(\boldsymbol{\theta}_i)$, where $F_i(\cdot)$ and $f_i(\cdot)$ are client i's local objective and empirical risk, respectively, and μ is a hyperparameter. By this means, the clients are no longer limited to implicitly acquiring the collaborative knowledge in an embedded (from the aggregated model weights) form.

The motivation behind this explicit design is that it facilitates the generalizability of the personalized models by directly penalizing their performance over other clients' risks, contributing to more informative updates and therefore better local performance. On the other hand, an implicit personalization scheme embeds the non-local information into the aggregated model weights, preventing the clients from directly engaging with other clients' risks.

We further demonstrate this motivation and the benefit of the explicit design by an empirical study: we personalize the output global model of FedAvg through $S = 6$ steps of local gradient-based update, supposing that the $O(SN^2)$ communication cost is affordable for now. We compare the exemplar explicit design of each client's local objective with a simple implicit method, i.e. the local fine-tuning of FedAvg where each client's local objective is only its own empirical risk ($F_i(\boldsymbol{\theta}_i) = f_i(\boldsymbol{\theta}_i)$).

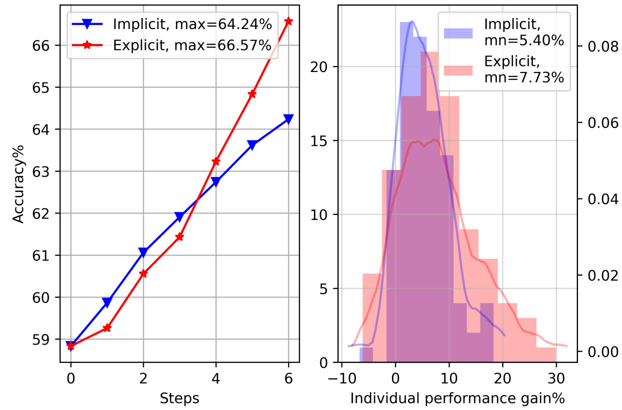


Figure 2. In the task of personalizing the output of FedAvg on CIFAR10 with 100 heterogeneous clients, the performance of the exemplar explicit local objective and an implicit local objective, assuming the communication cost is affordable. Figure (a) and (b) show the trend of the mean personalized test accuracy, and the histogram and density estimation of the individual gain, respectively.

The result of this empirical study are shown in Fig. 2. On CIFAR10 dataset with 100 heterogeneous clients, we observe that the explicit transfer of collaborative knowledge exhibits a stronger ability to adapt the model towards clients' local data than the implicit counterpart, with a mean individual performance gain of 7.73% over local test data from the initial global model, 2.33% higher than that of an implicit personalization. *Intriguing as the results may seem, how can we transfer this idea back to the communication-expensive real-world FL settings where acquiring all* $f_j(\boldsymbol{\theta}_i)\ \forall i,j \in [N]$ *will cost* $O(N^2)$ *communication overhead?* Our solution is to estimate $f_j(\boldsymbol{\theta}_i)$ by approximation as shown in Fig. 1.

Based on the above observation, in this work, we propose **P**ersonalized **G**lobal **Fed**erated Learning (PGFed), a novel personalized FL framework that enables each client to **personalize** its own **global** objective by *explicitly and adaptively* aggregating the empirical risks of itself and other clients. To avoid massive communication overhead and potential privacy leakage, each client's risk is estimated through a first-order approximation for other clients' adaptive risk aggregation. Thereby, the clients are able to explicitly acquire the collaborative knowledge, and their personalized models can enjoy better generalizability. We summarize our contributions as follows:

- We uncover that the explicitness of a personalized FL algorithm empowers itself with stronger adaptation ability. Based on this observation, we propose PGFed, a novel explicit personalized FL algorithm that frames the local objective of each client as a personalized global objective.
- To the best of our knowledge, PGFed is the first work

in the field to achieve explicit transfer of global collaborative knowledge among the clients, without introducing the seemingly unavoidable $O(N^2)$ communication costs.

- On top of PGFed, we develop a momentum upgrade, dubbed PGFedMo, to let the clients more efficiently utilize other clients' empirical risks.
- We evaluate PGFed and PGFedMo on four datasets under different FL settings. The results show that both algorithms outperform the compared state-of-the-art personalized FL methods, with up to **15.47% boost in accuracy**.

2. Related Work

Federated Learning. Federated learning [14, 21, 39] focuses on training a global consensus model over a federation of clients with similar data. The most notable FL algorithm, FedAvg [26], broadcast copies of the global model for local training and aggregate the updated copies as the new global model for the next round.

However, evidence began to accumulate in recent years that FedAvg is vulnerable towards data heterogeneity (non-IID data) [22, 32, 30, 13], which is the most common data distribution scenario in real-world FL. This vulnerability manifests itself as non-guaranteed convergence [30] and poor generalizability on test data after fine-tuning [13], which can cost the incentives of participating clients.

To improve the performance of the global model, FedProx [22] adds a proximal term to clients' empirical risks to restrict the local update. FedDyn [1] introduces a dynamic regularizer to each client to align the global and local optima. SCAFFOLD [15] corrects the local update by a control variate. And FedAlign [27] focuses on local model generality rather than proximal restrictions. However, while effective, these global FL algorithms cannot systematically alleviate the data heterogeneity challenge.

Personalized Federated Learning. Personalized FL [36, 18] allows different models for different clients. This relaxation fundamentally mitigates the impact caused by the non-IID data. Methods in this category are the ones we mainly compare with in our experiments.

In this branch of work, [2, 23, 5] focus on aggregating different layers of the model. A recent work, FedBABU [28], aggregates the feature extractor for a global model, and fine-tunes it for personalized models. Moreover, some works leverage multi-task [44] or meta-learning [10] to learn the relationships between clients' data distributions [34, 7, 3]. For instance, Per-FedAvg [7] takes advantage of Model-Agnostic Meta-Learning (MAML) [8] and trains an easy-to-adapt initial shared model. Instead of training a different model per client, clustered FL [9, 33] trains a distinct model for each cluster of similar clients. Interpolation of personalized models is also a popular direction [43, 25, 6], where works concentrate more on a personalized way to aggregate the models from different clients.

In addition, some recent works [43, 25] afford to pay the $O(N^2)$ communication cost to transmit the clients' models to other clients, but their massive communication cost only benefits the model aggregation, instead of paying more attention on aggregating clients' risks, which makes their algorithms fall into the implicit category. Another recent work [4] bridges the personalized FL and global FL by training a global model with balanced risk and personalized adaptive predictors with empirical risk. However, it is still implicit, as the clients do not engage with others' risks. In our work, we focus on an explicit way of personalization by introducing estimates of non-local empirical risks to each client without massive communication costs.

3. Problem Formulation

In this section, we formalize the problem of conventional and personalized FL, and further elaborate on the explicitness of personalized FL.

3.1. Conventional and Personalized FL

Conventional federated learning aims to train a global consensus model for a federation of clients with similar data. Take the most notable FL algorithm, FedAvg [26], as an example. For a federation of N clients, the goal of FedAvg is to minimize the global objective defined as:

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) = \sum_{i=1}^{N} p_i F_i(\boldsymbol{\theta}), \tag{1}$$

where $\boldsymbol{\theta}$ denotes the global model, $F_i(\cdot)$ represents the local objective of client i, and the weight p_i is often set as $p_i = n_i/n$ with $n = \sum_k n_k$ where n_k denotes the number of data samples on client k. In FedAvg, the local objective $F_i(\cdot)$ measures client i's empirical risk, i.e.

$$F_i(\boldsymbol{\theta}) = \mathbb{E}_{\xi \sim \mathcal{D}_i} f_i(\boldsymbol{\theta}|\xi) \approx \sum_{k=1}^{n_i} f_i(\boldsymbol{\theta}|\xi_k), \tag{2}$$

where $\mathcal{D}_i$ represents the data distribution, ξ_k is the k-th data sample on client i, and the empirical risk $f_i(\cdot)$ is used to approximate the true risk on client i. To simplify the notations, in the rest of the paper, we drop the summation and denote $f_i(\cdot)$ directly as $\sum_{k=1}^{n_i} f_i(\cdot|\xi_k)$ unless further clarified.

Personalized FL relaxes the number of models. Compared to conventional FL, each client i is allowed to have its own personalized model $\boldsymbol{\theta}_i$, i.e. the goal of personalized FL is defined as:

$$\min_{\boldsymbol{\Theta}} F(\boldsymbol{\Theta}) = \min_{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_N} \sum_{i=1}^{N} p_i F_i(\boldsymbol{\theta}_i), \tag{3}$$

where $\mathbf{\Theta}$ is a $d \times N$ matrix with d being the number of dimensions of the model.

In addition, the number of participating clients (N) in FL is often large [14], and not all clients are able to participate in each federated round. Therefore, a subset of clients $\mathcal{S}$ with $|\mathcal{S}| = M$ is selected for each round of training.

3.2. Implicit vs. Explicit Local Objective

As discussed in Sec. 1, we defined two types of ways for transferring collaborative knowledge from the server to the clients for personalized FL algorithms. The explicit way updates the model with a direct engagement with multiple clients' empirical risks. For instance, updating the global model of `FedAvg` achieves explicitness through communication. And in the empirical study (Fig. 2) in Sec. 1, we provided an example of explicit local objective as:

$$F_i(\boldsymbol{\theta}_i) = f_i(\boldsymbol{\theta}_i) + \frac{\mu}{(N-1)} \sum_{j \neq i} f_j(\boldsymbol{\theta}_i). \tag{4}$$

It is obvious that engaging all $f_j(\boldsymbol{\theta}_i)\ \forall i, j \in [N]$ to update the clients' personalized models would take $O(N^2)$ communication overhead. Therefore, most personalized FL algorithms implicitly transfer the collaborative knowledge by embedding it into the aggregation of model weights or as different regularizers.

One pitfall is that to see whether a personalized FL algorithm manages to afford the $O(N^2)$ communication overhead is *NOT* the criterion of checking its explicitness. [43, 25] are two recent works that did afford this communication cost on small federations, but since their personalized model updates do not directly involve other clients' risks, both of these algorithms still fall into the implicit category.

In addition, note that the explicitness of a personalized FL algorithm does not prevent itself from simultaneously possessing characteristics from the implicit counterpart. For instance, in a personalized FL algorithm, taking the globally aggregated model as a round-beginning initialization for clients' personalized models is an implicit characteristic that can co-exist with using Eq. (4) as an explicit personalization.

4. Method

In this section, we introduce the proposed algorithm, `PGFed`, in details. We clarify our global and local objectives, explain how we circumvent the massive communication cost, and introduce a momentum version of the algorithm, dubbed `PGFedMo`. The full procedure of `PGFed` is provided in Algorithm 1.

4.1. Objectives of `PGFed`

We adopt Eq. (3) as the global objective of `PGFed`. As mentioned in Sec. 3.2, an algorithm can simultaneously leverage characteristics from both explicit and implicit collaborative knowledge transfer. In each training round, we first implicitly transfer the collaborative knowledge to the clients by embedding it into the globally aggregated model, $\boldsymbol{\theta}_{glob}$, which serves as the round-beginning initialization for the clients' personalized models.

To address the explicitness, we design the local objective as a *personalized global objective* in the same form as the global objective of `FedAvg` (weighted sum of every clients' risks). Different from the previous exemplar design shown in Eq. (4), in `PGFed`, the local objective is defined as:

$$F_i(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i) = f_i(\boldsymbol{\theta}_i) + \mu \sum_{j \in [N]} \alpha_{ij} f_j(\boldsymbol{\theta}_i), \tag{5}$$

where $\boldsymbol{\alpha}_i$ is a N-dimensional vector of **learnable** scalars $\alpha_{ij} > 0$ which denotes the personalized weights of $f_j(\cdot)$ on client i. In this way, each client is able to personalize how much other clients' risks should contribute to their own objective. Since M clients are selected for every round, we initialize every as $\alpha_{ij} = 1/M\ \forall i, j \in [N]$. This new local objective slightly changes the global objective to:

$$\min_{\mathbf{\Theta}, \boldsymbol{A}} F(\mathbf{\Theta}, \boldsymbol{A}) = \min_{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N} \sum_{i=1}^{N} p_i F_i(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i), \tag{6}$$

where $\boldsymbol{A}$ is an $N \times N$ matrix with the i-th row being $\boldsymbol{\alpha}_i$.

Now, the question becomes: how can we overcome the massive communication cost to achieve explicit personalization? We answer this question in the next subsection.

4.2. Non-local Risk Estimation

In most real-world FL settings, $O(N^2)$ communication cost is often not affordable, and `PGFed` is no exception. To achieve the explicitness without having to pay the unaffordable cost, we estimate the non-local empirical risks by first-order approximations. Specifically, we define the non-local empirical risk on client i as $f_j(\boldsymbol{\theta}_i)\ \forall j \neq i$, and estimate this term by using Taylor expansion at $\boldsymbol{\theta}_j$, i.e.

$$f_j(\boldsymbol{\theta}_i) \approx f_j(\boldsymbol{\theta}_j) + \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j)^T (\boldsymbol{\theta}_i - \boldsymbol{\theta}_j), \tag{7}$$

where the higher-order terms are ignored. By plugging the approximation into Eq. (5), we have:

$$F_i(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i) \approx f_i(\boldsymbol{\theta}_i) + \mathcal{R}_{aux}^{[N]}(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i), \tag{8}$$

$$\mathcal{R}_{aux}^{[N]}(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i) = \mu \textstyle\sum_{j \in [N]} \alpha_{ij} \left(f_j(\boldsymbol{\theta}_j) + \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j)^T (\boldsymbol{\theta}_i - \boldsymbol{\theta}_j) \right). \tag{9}$$

where we define $\mathcal{R}_{aux}^{[N]}(\cdot)$ as an *auxiliary risk* over all client $j \in [N]$. It is the existence of the auxiliary risk $\mathcal{R}_{aux}^{[N]}(\cdot)$ that makes the proposed algorithm an explicit personalization.

The intuition behind why the approximation might work lies in the regularization effect of our explicit personalization. As we personalize different clients' models in the same form of global objective (Eq. (5)), the non-local risks restrain the personalized model weights from ungoverned drifting. Compared to an implicit setting (e.g. FedAvg [26]), the explicitly personalized models are more regularized, which enables the clients to train their personalized models in a more uniform direction. Therefore, as the regularization effect reduces the gaps between the personalized models, the first-order approximation is rewarded with a chance to shine in the proposed explicit personalization.

4.3. Gradient-based Update

In PGFed, the personalized models are updated through gradient-based optimizers such as stochastic gradient descent (SGD). In this section, we derive the gradient of the local objective of client i with respect to the personalized model $\boldsymbol{\theta}_i$ and the personalized weights $\boldsymbol{\alpha}_i$ for the non-local risks. Note that the gradient of the local objective is just the gradient of the local empirical risk plus the gradient of the auxiliary risk. Therefore, for the gradient w.r.t. $\boldsymbol{\theta}_i$, we have:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_i} F_i(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i) &= \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i) + \nabla_{\boldsymbol{\theta}_i} \mathcal{R}_{aux}^{[N]}(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i) \\ &= \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i) + \mu \underbrace{\sum_{j\in[N]} \alpha_{ij} \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j)}_{\tilde{\boldsymbol{g}}_{[N]}}. \end{aligned} \tag{10}$$

Since the $\tilde{\boldsymbol{g}}_{[N]}$, the *auxiliary gradient*, is not related to $\boldsymbol{\theta}_i$, we can have it computed by the server by acquiring $\boldsymbol{\alpha}_i$ from client i, and $\nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j)$ from client j.

For the gradient w.r.t. $\boldsymbol{\alpha}_i$, similarly, we have:

$$\begin{aligned}&\nabla_{\alpha_{ij}} F_i(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i) = \mu \left(f_j(\boldsymbol{\theta}_j) + \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j)^T (\boldsymbol{\theta}_i - \boldsymbol{\theta}_j) \right) \\ &= \underbrace{\mu \left(f_j(\boldsymbol{\theta}_j) - \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j)^T \boldsymbol{\theta}_j \right)}_{g_\alpha^{(1)}} + \underbrace{\mu \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j)^T \boldsymbol{\theta}_i}_{g_\alpha^{(2)}}. \end{aligned} \tag{11}$$

As shown in Eq. (11), this gradient can be split into two components: the first term, $g_\alpha^{(1)}$, is a scalar purely associated with client j, which can be uploaded from client j with little cost; the second term is an interaction between the gradient of client j and the personalized model of client i. To accurately acquire $g_\alpha^{(2)}$ term itself would, again, cost $O(N^2)$ communication overhead. Therefore, we approximate this term by an average, $\bar{\boldsymbol{g}}_{[N]}$, computed by the server, i.e.

$$g_\alpha^{(2)} \approx \bar{\boldsymbol{g}}_{[N]}^T \boldsymbol{\theta}_i = \frac{\mu}{N} \left(\sum_{j\in[N]} \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j) \right)^T \boldsymbol{\theta}_i. \tag{12}$$

Although transmitting $\bar{\boldsymbol{g}}_{[N]}$ can be costly, and it is possible for the server to transmit the scalar value of $g_\alpha^{(2)} \approx \bar{\boldsymbol{g}}_{[N]}^T \boldsymbol{\theta}_i$,

---

Algorithm 1 PGFed and PGFedMo

Input: N clients, learning rates η_1, η_2, number of rounds T, coefficient μ(, momentum β for PGFedMo)

Output: Personalized models $\boldsymbol{\theta}_1^T, ..., \boldsymbol{\theta}_N^T$.

ServerExecute:

Initialize $\alpha_{ij} \leftarrow 1/M\ \forall i, j \in [N]$, global model $\boldsymbol{\theta}_{glob}^0$
$\mathbf{A}[i] \leftarrow \boldsymbol{\alpha}_i\ \forall i \in [N]$
for $t \leftarrow 1, 2, ..., T$ **do**
  Select a subset of M clients, $\mathcal{S}_t$
  $g_t^{(1)} \leftarrow \{\}; \nabla_t \leftarrow \{\}$ // built for next round
  for $i \in \mathcal{S}_t$ **in parallel do**
    if t=1 **then**
      $\boldsymbol{\theta}_i^t, g_\alpha^{(1)}, \nabla f(\boldsymbol{\theta}_i^t), \boldsymbol{\alpha}_i \leftarrow$ **ClientUpdate**($\boldsymbol{\theta}_{glob}^{t-1}, t$)
    else
      $\tilde{\boldsymbol{g}}_{\mathcal{S}_{t-1}} \leftarrow \mu \sum_{j\in\mathcal{S}_{t-1}} \alpha_{ij} \nabla_{t-1}[j]$
      $\bar{\boldsymbol{g}}_{\mathcal{S}_{t-1}} \leftarrow \frac{\mu}{M} \sum_{j\in\mathcal{S}_{t-1}} \nabla_{t-1}[j]$
      $\boldsymbol{\theta}_i^t, g_\alpha^{(1)}, \nabla f(\boldsymbol{\theta}_i^t), \boldsymbol{\alpha}_i \leftarrow$ **ClientUpdate**($\boldsymbol{\theta}_{glob}^{t-1}, t, \tilde{\boldsymbol{g}}_{\mathcal{S}_{t-1}}, \bar{\boldsymbol{g}}_{\mathcal{S}_{t-1}}, g_{t-1}^{(1)}$)
    end if
    // the next line records the values for next round
    $\mathbf{A}[i] \leftarrow \boldsymbol{\alpha}_i; g_t^{(1)}[i] \leftarrow g_\alpha^{(1)}; \nabla_t[i] \leftarrow \nabla f(\boldsymbol{\theta}_i^t)$
    $\boldsymbol{\theta}_{glob}^t \leftarrow \sum_{i\in\mathcal{S}_t} p_i \boldsymbol{\theta}_i^t$
  end for
  for $i \in ([N] - \mathcal{S}_t)$ **in parallel do**
    $\boldsymbol{\theta}_i^t \leftarrow \boldsymbol{\theta}_i^{t-1}; \tilde{\boldsymbol{g}}_i^t \leftarrow \tilde{\boldsymbol{g}}_i^{t-1}$
  end for
end for
return $\boldsymbol{\theta}_1^T, ..., \boldsymbol{\theta}_N^T$

ClientUpdate($\boldsymbol{\theta}_{global}^{t-1}, t$ (, $\tilde{\boldsymbol{g}}, \bar{\boldsymbol{g}}, g_{t-1}^{(1)}$)):

if t=1 **then**
  $\boldsymbol{\theta}_i^t \leftarrow$ **ClientUpdate**($\boldsymbol{\theta}_{global}^{t-1}, \eta_1$) as in FedAvg
else
  $\boldsymbol{\theta}_i^t \leftarrow \boldsymbol{\theta}_{global}^{t-1}$
  $\tilde{\boldsymbol{g}}_i^t \leftarrow \tilde{\boldsymbol{g}}$ // without momentum
  $\tilde{\boldsymbol{g}}_i^t \leftarrow (1-\beta)\tilde{\boldsymbol{g}} + \beta \tilde{\boldsymbol{g}}_i^{t-1}$ // with momentum
  for Batch of data $\mathcal{B} \in \mathcal{D}_i$ **do**
    $\boldsymbol{\theta}_i^t \leftarrow \boldsymbol{\theta}_i^t - \eta_1(\nabla f(\boldsymbol{\theta}_i^t, \mathcal{B}) + \tilde{\boldsymbol{g}}_t^i)$
    $g^{(2)} = \bar{\boldsymbol{g}}^T \boldsymbol{\theta}_i$
    $\forall j \in g_{t-1}^{(1)} : \alpha_{ij} \leftarrow \alpha_{ij} - \eta_2(g_{t-1}^{(1)}[j] + g^{(2)})$
  end for
end if
$g_\alpha^{(1)} \leftarrow \mu \left(f(\boldsymbol{\theta}_i^t) - \nabla f(\boldsymbol{\theta}_i^t)^T \boldsymbol{\theta}_i^t \right)$ // for next round
return $\boldsymbol{\theta}_i^t, g_\alpha^{(1)}, \nabla f(\boldsymbol{\theta}_i^t), \boldsymbol{\alpha}_i$

---

we argue that it is not ideal to directly send this scalar to client i, because the calculations of $g_\alpha^{(2)}$ and $\nabla_{\alpha_{ij}} F_i(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i)$ are during the process of updating $\boldsymbol{\theta}_i$. And since calculating $g_\alpha^{(2)}$ involves $\boldsymbol{\theta}_i$, it would be more reasonable to treat $\bar{\boldsymbol{g}}_{[N]}^T \boldsymbol{\theta}_i$

	CIFAR10			CIFAR100		
	25 clients	50 clients	100 clients	25 clients	50 clients	100 clients
Local	72.40 ± 0.45	70.28 ± 0.38	67.39 ± 0.20	32.74 ± 0.08	26.05 ± 0.34	23.06 ± 0.47
FedAvg [26]	65.07 ± 0.25	64.41 ± 0.66	63.19 ± 0.46	28.48 ± 0.59	26.06 ± 0.65	25.58 ± 0.80
FedDyn [1]	67.31 ± 0.36	65.02 ± 0.91	62.49 ± 0.06	34.17 ± 0.43	27.06 ± 0.18	23.88 ± 0.36
pFedMe [35]	70.60 ± 0.23	68.92 ± 0.35	66.40 ± 0.04	27.97 ± 0.24	23.82 ± 0.06	22.35 ± 0.03
FedFomo [43]	72.33 ± 0.03	72.17 ± 0.48	70.86 ± 0.27	32.15 ± 0.61	25.90 ± 1.17	24.48 ± 0.44
APFL [6]	77.03 ± 0.26	77.36 ± 0.18	76.29 ± 0.13	39.16 ± 0.93	35.15 ± 0.65	33.86 ± 0.60
FedRep [5]	76.85 ± 0.44	76.03 ± 0.17	72.30 ± 0.52	33.43 ± 0.80	26.86 ± 0.39	22.76 ± 0.45
LG-FedAvg [23]	72.83 ± 0.28	70.44 ± 0.31	67.55 ± 0.09	33.65 ± 0.19	27.13 ± 0.37	24.82 ± 0.28
FedPer [2]	77.84 ± 0.18	77.76 ± 0.22	75.01 ± 0.20	35.22 ± 0.67	28.63 ± 0.70	25.56 ± 0.26
Per-FedAvg [7]	75.49 ± 0.74	76.27 ± 0.50	75.41 ± 0.35	32.89 ± 0.43	32.24 ± 0.75	32.59 ± 0.21
FedRoD [4]	79.73 ± 0.68	79.61 ± 0.22	77.76 ± 0.32	39.55 ± 0.58	33.87 ± 2.42	31.49 ± 0.19
FedBABU [28]	78.92 ± 0.36	79.35 ± 0.84	76.34 ± 0.22	32.71 ± 0.23	29.66 ± 0.64	27.72 ± 0.11
PGFed (ours)	<u>81.02 ± 0.41</u>	<u>81.42 ± 0.31</u>	<u>78.56 ± 0.35</u>	<u>43.12 ± 0.03</u>	<u>38.45 ± 0.44</u>	<u>35.71 ± 0.54</u>
PGFedMo (ours)	**81.20 ± 0.08**	**81.48 ± 0.32**	**78.74 ± 0.22**	**43.44 ± 0.14**	**38.50 ± 0.45**	**35.76 ± 0.65**

Table 1. Mean top-1 personalized accuracy of the proposed algorithms and the baselines. We report the mean and standard deviation over three different seeds. The highest and second-highest accuracies under each setting are in **bold** and <u>underlined</u>, respectively. **Within the comparison of personalized FL algorithms, PGFed and PGFedMo boost the accuracy by up to 15.47%.**

as a variable and have it computed by client i locally, rather than treating it as a constant computed it by the server.

In the descriptions of the proposed algorithm above, we did not elaborate on how the client selection procedure would affect the auxiliary risks of each client. As mentioned in Sec. 4.1, in each round t, a subset of clients, $\mathcal{S}_t$ with $|\mathcal{S}_t| = M$, are selected for the training. We, therefore, slightly modify the auxiliary risk from $\mathcal{R}_{aux}^{[N]}$ to $\mathcal{R}_{aux}^{\mathcal{S}_t}$. This change will subsequently affect the scope of two terms, namely the auxiliary gradient $\tilde{\boldsymbol{g}}_{[N]}$ w.r.t. $\boldsymbol{\theta}_i$ (changed to $\tilde{\boldsymbol{g}}_{\mathcal{S}_t}$), and $\bar{\boldsymbol{g}}_{[N]}$, a portion of the gradient w.r.t. α_{ij} (changed to $\bar{\boldsymbol{g}}_{\mathcal{S}_t}$). We formally define these two new terms below:

$$\tilde{\boldsymbol{g}}_{\mathcal{S}_t} = \mu \sum_{j \in \mathcal{S}_t} \alpha_{ij} \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j), \tag{13}$$

$$\bar{\boldsymbol{g}}_{\mathcal{S}_t} = \frac{\mu}{M} \sum_{j \in \mathcal{S}_t} \nabla_{\boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j). \tag{14}$$

To sum up, the auxiliary risk $\mathcal{R}_{aux}^{\mathcal{S}_t}$ on a client ultimately results in downloading two server-aggregated gradients $\tilde{\boldsymbol{g}}_{\mathcal{S}_t}$ and $\bar{\boldsymbol{g}}_{\mathcal{S}_t}$ (and a negligible scalar $g_\alpha^{(1)}$). Note that while achieving the explicitness, this server aggregation makes it infeasible for the client to separate other clients' gradient for inferring their data, which protects the clients' privacy.

4.4. PGFed with Momentum (PGFedMo)

When considering the client selection for each round, the auxiliary risk changed from $\mathcal{R}_{aux}^{[N]}$ to $\mathcal{R}_{aux}^{\mathcal{S}_t}$. This reduces the number of clients' risks involved in the calculation of auxiliary risk from N to M, which could be a huge loss of information. To compensate for this loss, each client maintains $\tilde{\boldsymbol{g}}_{\mathcal{S}_t}^i$ locally, and every time when the client is selected, it updates its local auxiliary gradient $\tilde{\boldsymbol{g}}_{\mathcal{S}_t}^i$ in a momentum manner, i.e.

$$\tilde{\boldsymbol{g}}_{\mathcal{S}_t}^i \leftarrow (1-\beta)\tilde{\boldsymbol{g}}_{\mathcal{S}_t}^i(\text{downloaded}) + \beta\tilde{\boldsymbol{g}}_{\mathcal{S}_t}^i(\text{previous}) \tag{15}$$

In this way, the clients are able to carry the auxiliary gradient, without having to discard the collaborative knowledge from the $N - M$ clients each round. We summarize the proposed algorithm in Algorithm 1.

5. Experiments

5.1. Experimental Setup

Compared methods. We evaluate the proposed algorithms, PGFed and PGFedMo, against a number of state-of-the-art FL algorithms, including two global FL algorithms: the leading FedAvg and FedDyn that uses a dynamically regularized local training, and nine personalized FL algorithms: pFedME which conducts personalization with Moreau Envelopes; FedFomo and APFL which interpolates personalized models by interpolation; FedRep, LG-FedAvg, and FedPer which personalize different layers of the model; Per-FedAvg which leverages meta-learning to learn an initial shared model, and one-step finetuning over local data; FedRoD which trains a global model with balanced risk and personalized adaptive predictors with empirical risk; and FedBABU which focuses on global representation learning and personalize the model by finetuning for one epoch.

Datasets. We conduct experiments on four datasets. In the main content, we report the results on CIFAR10 and CIFAR100 [17], and we leave the experiments on OrganAMNIST [42] and Office-home [37] dataset in the **supplementary materials**. Following [11], we use Dirichlet distribution with $\alpha = 0.3$ to partition the dataset into heterogeneous settings with 25, 50, and 100 clients. For each setting, each client's local training and test datasets are under the same distribution. We report the mean top-1 personalized accuracy by averaging the local test accuracies over all clients.

Implementation details. We adopt the same convolutional neural network (CNN)'s architecture as in `FedAvg` for all the compared methods. There are 2 convolutional layers with 32, and 64 5×5 kernels and 2 fully connected layers with 512 hidden units in this architecture. For the methods that involve representation learning, we split the model into a feature extractor with the first three layers, and a classifier with the last layer. We use a stochastic gradient descent (SGD) optimizer for every method with a fixed momentum of 0.9. Each method is trained on the dataset for 300 federated rounds. For each federated setting, the client sample rate is set to 25%, and the local training epochs is set to 5. The learning rate is tuned from {0.1, 0.01, 0.001, 0.0001} for every method. For `PGFed`, the coefficient for the auxiliary risk, μ, is tuned from {0.1, 0.05, 0.01, 0.005, 0.001}. For `PGFedMo`, the momentum, β, is tuned from {0.2, 0.5, 0.8}. More details about hyperparameters values are included in the *Supplementary Material*.

5.2. Main Results and Analysis

We report the mean and standard deviation of three different seeds for each setting. As shown in Tab. 1, our method achieves the highest mean accuracies under all settings. With large data heterogeneity using Dir(0.3) as in our experiments, local training can often achieve better performance than participating in global FL. As the number of clients increases, the clients benefit more from FL, since local training is more likely to overfit on the small local datasets. **Within the comparison of personalized FL algorithms, `PGFed` and the momentum upgrade, `PGFedMo`, enjoy performance improvements by up to 15.47% in accuracy (CIFAR100 with 25 clients)**, outperforming the methods that personalize different components of the model such as `FedRep`, `LG-FedAvg`, and `FedPer`, and even with the fine-tuning in `FedBABU` and the meta-learning `Per-FedAvg`, as well as the interpolation methods such as `APFL` and `FedFomo`. With the global model and the personalized adapter, `FedRod` benefits the local performance by the most within the baselines, yet was still exceeded by `PGFed` and `PGFedMo` by up to 4.63% (on CIFAR100 with 50 clients). For `PGFed`, the momentum upgrade, `PGFedMo`, further boosts its performance since more clients' risk is included in the clients' auxiliary risks with the momentum update.

	25 clients		50 clients		100 clients	
	#round	speedup	#round	speedup	#round	speedup
`APFL`	31	1.0×	28	1.7×	24	2.6×
`FedPer`	8	3.9×	6	7.8×	8	7.9×
`Per-FedAvg`	31	1.0×	47	1.0×	63	1.0×
`FedRoD`	26	1.2×	35	1.3×	10	6.3×
`PGFed`	9	3.4×	14	3.4×	15	4.2×
`PGFedMo`	9	3.4×	14	3.4×	15	4.2×

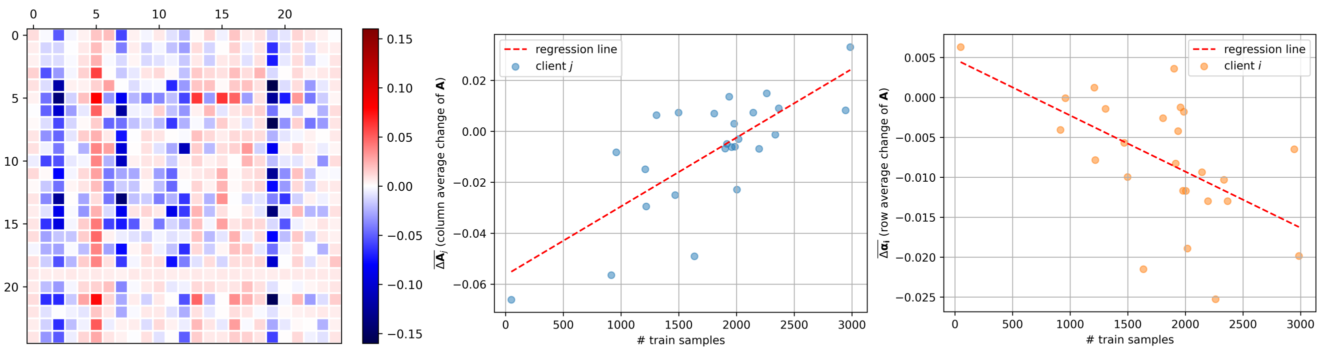
Table 2. The number of rounds to achieve 70% mean top-1 personalized accuracy on CIFAR10. The speedup is computed based on the slowest approach listed (i.e. "1.0×").

In Tab. 2, we show the convergence speed of the personalized algorithms by reporting the round numbers at which the algorithms achieve 70% mean top-1 personalized accuracy on CIFAR10. For each setting, we set the algorithm that takes the most round to reach 70% as "1.0×", and find that the proposed `PGFed` and `PGFedMo` consistently present a decent amount of speedup from the compared personalized FL algorithms. Among the six compared personalized FL algorithms, `FedPer` has the fastest convergence speed, due to the smaller size of the globally aggregated model component (only the feature extractor layers are aggregated in `FedPer`). Although not fastest, `PGFed` and `PGFedMo` still have an average of 3.7× speed up in convergence rate, while enjoying the best performance in accuracy.

	25 clients	50 clients	100 clients
`FedAvg`	-8.99 ± 10.36	-8.90 ± 15.48	-5.02 ± 14.30
`APFL`	2.79 ± 8.07	5.73 ± 8.43	8.37 ± 6.91
`FedPer`	5.31 ± 2.56	8.31 ± 6.00	8.63 ± 5.26
`Per-FedAvg`	0.72 ± 6.22	5.02 ± 7.39	8.09 ± 7.00
`FedRoD`	7.80 ± 3.68	8.84 ± 6.29	10.68 ± 6.14
`PGFed`	8.49 ± 4.67	10.78 ± 5.88	11.15 ± 5.06
`PGFedMo`	8.61 ± 3.59	10.90 ± 6.11	11.16 ± 5.44

Table 3. Mean and standard deviation of the individual performance gain over `Local` training in terms of accuracy% on local test set on CIFAR10.

Besides the overall performance and the convergence speed, we take a micro-perspective to examine the individual performance gain over the `Local` training. Specifically, we concentrate on the statistics of the individual performance gain across the federation of clients, which can indicate the fair performance of the algorithms. For instance, the individual gain with a high standard deviation might indicate that the performance gains on some clients come with a sacrifice over the performance gains (little or negative gains) on other clients. In Tab. 3 and 4, we show



(a) (b) (c)

Figure 3. Visualization of the change in $\boldsymbol{A}$. (a) is a heat map of the change in $\boldsymbol{A}$. For (b) and (c), the Y-axis of Figure (b) represents the column average of the change in $\boldsymbol{A}$ (the average change of weights of client j's empirical risk on other clients). The Y-axis of (c) is the row average of the change in $\boldsymbol{A}$ (the average change of weights of the auxiliary risk on client i). Through the regression line, we verify the positive correlation between $\overline{\Delta \boldsymbol{A}_j}$ and n_j in (b), and the negative correlation between $\overline{\Delta \boldsymbol{\alpha}_i}$ and n_i in (c).

	25 clients	50 clients	100 clients
FedAvg	-3.29 ± 4.22	0.02 ± 4.63	1.77 ± 6.38
APFL	6.48 ± 2.93	8.70 ± 3.37	9.31 ± 4.55
FedPer	3.43 ± 1.80	2.16 ± 2.45	2.31 ± 3.54
Per-FedAvg	0.07 ± 3.71	5.47 ± 3.86	7.49 ± 5.73
FedRoD	7.32 ± 2.68	6.59 ± 3.17	7.47 ± 3.69
PGFed	9.34 ± 1.71	9.01 ± 2.97	12.05 ± 3.93
PGFedMo	9.40 ± 1.87	8.99 ± 2.76	12.07 ± 3.97

Table 4. Mean and standard deviation of the individual performance gain over `Local` training in terms of accuracy% on local test set on CIFAR100.

that while enjoying the highest mean individual gains under each setting, the standard deviations of both `PGFed` and `PGFedMo` are reasonably small as well, suggesting a fair boost of performance from every client. We attribute the high mean and low standard deviation merit of the proposed algorithms to the explicitness of our design of local objectives. Although clients' local objectives are personalized, the explicitness enables the clients' local objectives to be of the same form (weighted sum of multiple empirical risks). And each client's local empirical risk takes roughly $1/(1+\mu)$ of its own objective, which is the same for every client, hence the similar individual performance gain and fair federated personalization.

5.3. Visualization of Coefficient Matrix $\boldsymbol{A}$

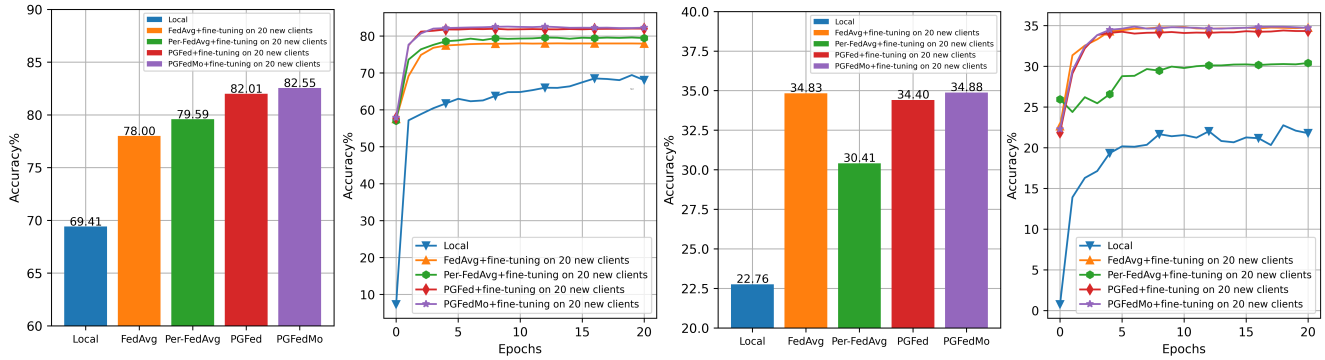
We investigate the coefficient matrix, $\boldsymbol{A}$, of α_{ij}'s in `PGFed` (recall that α_{ij} is the coefficient of client j's empirical risk on client i). Specifically, we visualize the ***change*** of all α_{ij} $\forall i, j \in [N]$ on CIFAR10 with 25 clients. As a refresher, $\boldsymbol{\alpha}_i$, the i-th row of $\boldsymbol{A}$, represents the vector of weights indicating how much client i values other clients' risks. The j-th column of $\boldsymbol{A}$, $\boldsymbol{A}_j$, represents the vector of weights indicating how much client j is valuable towards other clients' risks.

In theory, without the first-order estimation (see Eq. (5)), the gradient of α_{ij} equals $\mu f_j(\boldsymbol{\theta}_i)$, which is non-negative. This suggests that α_{ij} will always converge to 0 given enough time, i.e. $\forall i, j \in [N], \lim_{t\to\infty} \alpha_{ij} = 0$. However, since $f_j(\boldsymbol{\theta}_i)$ is estimated through Taylor expansion, and higher order terms are omitted, in practice of `PGFed`, α_{ij} has a chance to increase (pink and red blocks in Fig. 3(a)), or decreases more slowly (light blue in Fig. 3(a)), as long as client i's local objective decreases. Therefore, whether α_{ij} changes in the positive or negative directions depends on whether $f_j(\cdot)$ can be "helpful" to reduce the local objective of client i as a result of optimization.

We quantify this "help" that client j can offer by $\overline{\Delta \boldsymbol{A}_j}$: the average change of the j-th column of $\boldsymbol{A}$ (i.e. α_{ij} $\forall i \in [N]$), and quantify the "help" that client i needs by $\overline{\Delta \boldsymbol{\alpha}_i}$: the average change of the i-th row of $\boldsymbol{A}$ (i.e. α_{ij} $\forall j \in [N]$). An intuitive indicator of this "help" is the number of local training samples. From the perspective of client j (the helper), larger local training set of client j should be able to make its empirical risk more likely to be helpful to other clients. From the perspective of client i (the helpee), smaller training set of client i should require more help from others. Therefore, ideally, $\overline{\Delta \boldsymbol{A}_j}$ and $\overline{\Delta \boldsymbol{\alpha}_i}$ should be positively and negatively correlated to the number of local training samples, respectively. This is verified as a finding in the resulting $\boldsymbol{A}$ of `PGFed` in Fig. 3(b) and Fig. 3(c).

5.4. Generalizability to New Clients

In real-world FL settings, it is possible that some clients did not participate in the FL training, but wish to have a model that could quickly adapt to their local data. This makes it especially desirable for a personalized FL algorithm to possess excellent adaptive ability. By design, `PGFed` is able to generate a global model as a side product



(a) (b) (c) (d)

Figure 4. A comparison w.r.t. generalizability on new clients on CIFAR10 and CIFAR100. We fine-tune the global models of different FL approaches and compare them with `Local` training. For FL approaches, we first train the global models on 80 clients for 150 rounds before fine-tuning them on 20 new clients for 20 epochs. (a) and (c) show the final mean personalized accuracies over the 20 new clients on CIFAR10 and CIFAR100, respectively. (b) and (d) show the learning curve along the fine-tuning.

(line 16 of Algorithm 1). Therefore, we simulate such a setting and study the generalizability of our global model, and compare it against several baseline algorithms (global and personalized) that also produce the global model. Specifically, we simulate this setting by conducting FL over 80 clients randomly selected from 100 clients on CIFAR10 and CIFAR100 datasets, and fine-tuning the trained global model on the rest 20 new clients. Besides `Local` training, we compare `PGFed` and `PGFedMo` with `FedAvg` and `Per-FedAvg`, whose goal is to train an easy-to-adapt model. The results are shown in Fig. 4.

Since the local fine-tuning on the 20 new clients is a standard vanilla training with SGD for all five compared methods, a high mean personalized accuracy directly indicates stronger overall generalizability of the global model (for FL algorithms), especially when the starting mean accuracies (before fine-tuning) are roughly the same. From Fig. 4, we can see that the `Local` trained models do not generalize well on local test data, since the size of the local dataset is likely to be small with a 100-client setting. For the FL algorithms, `PGFedMo` achieves the best mean personalized accuracy, which shows strong generalizability of the global model of the proposed algorithm thanks to the explicit design. While not being the main focus, the high generalizability of `PGFed`'s and `PGFedMo`'s global models shown on new clients also indicates the models' strong adaptiveness in the original personalized FL task.

6. Conclusion and Discussion

In this work, we discovered that a personalized FL algorithm's explicitness enhances models' generalizability, resulting in better local performance. Based on our observations, we proposed, `PGFed`, and its momentum upgrade, `PGFedMo`. Both algorithms explicitly transfer the collaborative knowledge across the clients by formulating their local objectives as personalized global objectives. This is achieved without introducing the seemingly unavoidable massive communication costs or potential privacy risk. Our extensive experiments demonstrated the improvements of the proposed algorithms over state-of-the-art methods on four datasets under different heterogeneous FL settings.

We expect the proposed framework to be extended in different directions. First, since the proposed framework is agnostic, it can be potentially combined with existing implicit FL algorithms such as [23, 5, 6, 2, 4, 28]. Moreover, although the proposed work manages to avoid the $O(N^2)$ communication overhead, since it still costs roughly $2.5\times$ the communication of `FedAvg`, a more communication-efficient method is also worth investigations. We leave these directions for future work.

Acknowledgements

This work was supported in part by a National Institutes of Health (NIH) / National Cancer Institute (NCI) grant (1R01CA218405), a National Science Foundation (NSF) grant (CICI: SIVD: 2115082), the grant 1R01EB032896 as part of the NSF/NIH Smart Health and Biomedical Research in the Era of Artificial Intelligence and Advanced Data Science Program, a NIH R01 Supplement grant (3R01EB032896-03S1), a Pitt Momentum Funds scaling award (Pittsburgh Center for AI Innovation in Medical Imaging), and an Amazon AWS Machine Learning Research Award. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF grant number ACI-1548562. Specifically, it used the Bridges-2 system, which is supported by NSF award number ACI-1928147, at the Pittsburgh Supercomputing Center. This work was also partially supported by the NSF/Intel Partnership on MLWiNS under Grant No. 2003198.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2020. 1, 3, 6

[2] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019. 2, 3, 6, 9

[3] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018. 2, 3

[4] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2022. 2, 3, 6, 9

[5] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021. 1, 2, 3, 6, 9

[6] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020. 2, 3, 6, 9

[7] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020. 2, 3, 6

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 3

[9] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020. 2, 3

[10] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021. 2, 3

[11] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 7

[12] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7865–7873, 2021. 2

[13] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019. 1, 3

[14] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. 1, 3, 4

[15] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 1, 3

[16] Latif U Khan, Walid Saad, Zhu Han, Ekram Hossain, and Choong Seon Hong. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, 2021. 1

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7

[18] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. Survey of personalization techniques for federated learning. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 794–797. IEEE, 2020. 1, 3

[19] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019. 2

[20] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10713–10722, 2021. 1

[21] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. 1, 3

[22] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. 1, 3

[23] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020. 1, 2, 3, 6, 9

[24] Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. Federated learning for open banking. In *Federated learning*, pages 240–254. Springer, 2020. 1

[25] Jun Luo and Shandong Wu. Adapt to adaptation: Learning personalization for cross-silo federated learning. *arXiv preprint arXiv:2110.08394*, 2021. 2, 3, 4

[26] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 3, 5, 6

[27] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8397–8406, 2022. 3

[28] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021. 2, 3, 6, 9

[29] Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *International Conference on Machine Learning*, pages 17716–17758. PMLR, 2022. 1

[30] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 3:3, 2018. 1, 3

[31] Felix Sattler, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. On the byzantine robustness of clustered federated learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8861–8865. IEEE, 2020. 2

[32] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019. 1, 3

[33] Geet Shingi. A federated learning based approach for loan defaults prediction. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 362–368. IEEE, 2020. 2, 3

[34] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. *arXiv preprint arXiv:1705.10467*, 2017. 2, 3

[35] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020. 6

[36] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 1, 3

[37] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 7

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

[39] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021. 1, 3

[40] Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019. 2

[41] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021. 1

[42] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021. 7

[43] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*, 2020. 2, 3, 4, 6

[44] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017. 2, 3