Preserving Global Performance in Personalized Federated Learning

Anonymous ICCV submission

Paper ID 6989

Contents

A . Algorithm and Convergence Analysis

We first analyze the direction for the model update in Federated Learning (FL). We then provide further information about S-FedPG with the step size line search procedure. Finally, we demonstrate the proof of the convergence of S-FedPG and FedPG in Appendix A.3.

A.1 . Analysis of Direction for the Model Update

Our proposed FedPG mainly includes two steps:

- Enhance the global performance of the global model (G-model) for better building personalized models (P-models).
- Extend the G-model to build P-models for clients to achieve personalization while preserving global performance.

A.1.1 Direction for Global Model Update

Since the P-model is built on the guidance of the G-model, a G-model with good global performance can help better build P-models. In the main paper, we list three challenges to the performance improvement of the G-model: **(A) Gradients conflict, (B) Objective bias, and (c) Absent-client conflict.**

(1) Handle the first two challenges.

Denote d^t as the direction for updating the global model ω^t at t^{th} communication round in FL, then d^t can motivate the improvement of each local objective $L_i(\omega^t)$ when satisfying

$$g_i^t \cdot d^t < 0,\ \forall i \in 1, \cdots, m, \tag{1}$$

where $g_i^t = \nabla L_i(\omega^t)$ represents the local gradient of L_i with respect to ω^t.

Related Work. Wang et al. [19] show that d^t obtained by traditional FL methods cannot ensure that d^t can satisfy (1) in non-IID settings, which will lead to the conflict with some clients, i.e., $g_i^t \cdot d^t > 0$, and thus the objective of client i will be reduced and harm generalization. It's a proactive way to enhance the performance of FL under non-IID settings by calculating a non-conflicted direction to update the global model. However, just a few attempts have been made so far, and they have failed. Wang et al. [19] propose a fair FL algorithm FedFV [19] to mitigate the conflicts in FL. FedFV can eliminate this kind of conflict **when there are only two clients in FL.** However, if there are more than two clients, it cannot guarantee that the conflicts will not occur. In Fig.1 of the main paper, we show a counterexample of it. Hu et al. propose FedMGDA+ [9] that attempts to calculate a common descent direction d^t satisfying (1). Regrettably, it contains a hyper-parameter (denote μ here) that is hard to tune. For example, when μ is set to 0.1, which is frequently used in their paper, FedMGDA+ usually fails to obtain a common descent direction. We provide a counterexample in Fig.1 of the main paper.

Besides the gradient conflicts, we observe that **objective bias** is another challenge. Even though we utilize some techniques to calculate a common descent direction, the amount of descent for clients' objective may vary greatly and easily lead to a state where the G-model performs well in some specific clients while poorly on others. We gave a demo in Fig.1.(b) of the main paper.

To address the above two issues, we propose a novel multi-objective optimization problem for FL with a fair-driven objective $min\ F(\omega)$:

$$\min_{\omega}\ (L_1(\omega), L_2(\omega), \cdots, L_m(\omega), F(\omega)), \tag{2}$$

where $F(\omega) = -\frac{L(\omega)\cdot\vec{1}}{\|L(\omega)\|\|\vec{1}\|}$ represents the negative of the cosine similarity between $L(\omega)$ and $\vec{1}$. We calculate a direction that not only can lead to increase clients' objectives, but also can reduce $F(\omega)$ and thus mitigate the objective bias. We adopt the technique of MGDA to compute a direction that can not only drive $L_i(\omega), \forall i$ to be smaller, but also cause $F(\omega)$ to be smaller. We first solve the follow quadratic problem to obtain λ, and then compute d^t by $d^t = -Q\lambda$.

$$\begin{aligned} &\max_{\lambda} -\tfrac{1}{2}\lambda^T(Q^TQ)\lambda \\ &s.t.\ \ \textstyle\sum_{i=1}^{|\lambda|}\lambda_i = 1, \\ &\qquad \lambda_i \geq 0, \forall i = 1, 2, \cdots, |\lambda|, \end{aligned} \tag{3}$$

Note that Q^TQ is positive-semidefinite, we can always obtain the optimal λ of Problem (3). In the experiment, we utilize cvxopt [1] to compute λ. Inspired by [7], the obtained d^t satisfies:

[Lemma 1] If $d^t = -Q\lambda$ and λ is the optimal solution of Problem (3), then:
1. If ω^t is Pareto stationary, then $d^t = \vec{0}$.
2. If ω^t is not Pareto stationary, then

$$\begin{aligned} & g_i^t \cdot d^t < 0, \ \forall i = 1, \cdots, m, \\ & \nabla F(\omega^t) \cdot d^t < 0. \end{aligned} \tag{4}$$

Therefore, d^t can drive to reduce $L_i(\omega^t), \forall i$ and $F(\omega^t)$ when ω^t is not the Pareto stationary. Now we prove that if ω is a Pareto stationary point in Problem (2), it is also Pareto stationary in the following Problem:

$$\min_{\omega} \ (L_1(\omega), L_2(\omega), \cdots, L_m(\omega)). \tag{5}$$

Since $\nabla F(\omega^t) = \frac{g^t}{\|L(\omega^t)\|} \cdot (\frac{L(\omega^t)^T \vec{1} L(\omega^t)}{\|\vec{1}\| \|L(\omega^t)\|^2} - \frac{\vec{1}}{\|\vec{1}\|})$, denote $h = \frac{1}{\|L(\omega^t)\|} \cdot (\frac{L(\omega^t)^T \vec{1} L(\omega^t)}{\|\vec{1}\| \|L(\omega^t)\|^2} - \frac{\vec{1}}{\|\vec{1}\|})$, then we can see that h is a scaled vector lying in the projection of $\vec{1}$ to the normal plane of $L(\omega^t)$. Thus, $h \perp L(\omega^t)$.

When ω^t is not a Pareto stationary point in Problem (5), we have $L_i(\omega^t) > 0, \forall i$.

Hence, there is at least one element of the vector h that is bigger than 0.

Therefore, when ω^t is not Pareto stationary in Problem (5), there always exists a direction d^t satisfying $g^t \cdot h \cdot d^t < 0$, i.e., $\nabla F(\omega^t) \cdot d^t < 0$, meaning that it is also not a Pareto stationary point in Problem (2). Q.E.D.

(2) Mitigate absent-client conflict.

To handle the third challenge: absent-client conflict, we introduce an effective strategy in Section 3.2.2 of the main paper, which can mitigate the negative impact of partial client participation/dropout.

A.1.2 Direction for Personalized Models Update

Given that there is a trade-off between generalization and personalization, it is easy for the global performance of the personalized model (P-model) to suffer from reduction when the personalization is enhanced. Previous Personalized Federated Learning algorithms cannot achieve a robust balance between generalization and personalization since the global performance of the P-model undergoes uncontrollable degrees of reduction in different cases. When generating a P-model ω_i^t from the global model ω^t for client i, the reason for the generalization reduction is that the direction d_i^t for generating ω_i^t conflicts with the local gradient of other clients, i.e., $g_j^t \cdot d_i^t > 0, \exists j \neq i$. Thus, when client i gets data similar to other clients in the future, the accuracy of the P-model will be decreased.

In contrast, **the most significant contribution of our proposed FedPG is that it can compute a generalization-protected personalized direction to generate the P-model**. It does not compromise global performance when enhancing personalization, thereby achieving a robust trade-off between generalization and personalization in different scenarios.

A.2. Step Size Line Search

For the purpose of dynamically adjusting the step size η^t for the model update, we propose S-FedPG that replaces the manual fixed or simple decayed learning rate in FedPG with the one obtained by the step size line search. Algorithm 1 depicts the procedure of the step size line search, where the input parameters come from the main algorithm of FedPG mentioned in the main paper.

Line 6 to 14 in Algorithm 1 is the first trial for searching a proper step size. When the Armijo condition

$$\begin{aligned} & L_i(\omega^t) \geq L_i(\omega^t + \eta^t d^t) - \beta \eta^t g^t d^t, \forall i \in S^t \\ & F(\omega^t) \geq F(\omega^t + \eta^t d^t) - \beta \eta^t \nabla F(\omega^t) \cdot d^t, \end{aligned} \tag{6}$$

in the first trial cannot be satisfied, it initiates the second trial for searching a proper step size that satisfies a more relaxed condition

$$\frac{1}{m} \sum\nolimits_{i=1}^{m} L_i < \frac{1}{m} \sum\nolimits_{i=1}^{m} L_i(\omega^t), \tag{7}$$

which can guarantee the original FL objective $\frac{1}{m} \sum_{i=1}^{m} L_i(\omega^t)$ to be smaller. If the two trials of step size searching fail to determine a step size, it reflects that the learning rate η is set too large, since S-FedPG searches η^t in the range of $[\eta_{lb}, \eta_{ub}]$. σ is computed by $\|d_r\| / \|d^t\|$ in the main paper. Note that η_{ub} is set to η when $S^{t-1} \not\subset S^t$. This is because when there are clients absent, it is necessary to prevent the step size from being too large, which may harm the objective of the absent clients.

Algorithm 1 StepSizeLineSearch

```
**Input**: $s, S^t, S^{t-1}, \eta, L(\omega^t), g^t, \nabla F(\omega^t), d^t$
**Output**: $\eta^t$
$\eta_{ub} \leftarrow 2^s\eta$; ($\eta_{ub} \leftarrow \eta$ when $S^{t-1} \not\subset S^t$)
$\eta_{lb} \leftarrow (1/2)^s\eta/\sigma$.
Initialize $\eta^t \leftarrow \eta_{ub}$, and history set $H$.
$stop \leftarrow False$.
while $\eta^t \geq \eta_{lb}$ do
    Send $\eta^t$ to all clients in $S^t$.
    Each client $i$ sends $L_i = L_i(\omega^t + \eta^t d^t)$ to the server.
    $L \leftarrow (L_1, \cdots, L_{|S^t|})$.
    Store the pair $(\eta^t, L)$ in the history set $H$.
    if $L_i(\omega^t) \geq L_i - \beta\eta^t g^t d^t, \forall i \in S^t$ and
      $F(\omega^t) \geq F(\omega^t + \eta^t d^t) - \beta\eta^t \nabla F(\omega^t) \cdot d^t$ then
        $stop \leftarrow True$.
        break
    $\eta^t \leftarrow \eta^t/2$.
if $stop = False$ then
    Select the biggest $\eta^t$ in $H$ that the corresponding $L$ satisfies
    $\frac{1}{m}\sum_{i=1}^m L_i < \frac{1}{m}\sum_{i=1}^m L_i(\omega^t)$.
if No $L$ in $H$ satisfies the above stopping condition then
    Select $(\eta^t, L)$ from $H$ that $\frac{1}{m}\sum\limits_{i=1}^{m} L_i$ is the smallest.
```

A.3. Convergence Analysis

A.3.1 Convergence Analysis of S-FedPG

We give the proof of the convergence of S-FedPG as follows.

Assume that all clients are online at each round, and their objectives are differentiable and smooth. Denote the objectives function of all clients: $L : \mathbb{R}^n \to \mathbb{R}^m$.

Assume that the hyper-parameter s in S-FedPG is set large enough that η_{lb} can satisfy the Armijo condition, we have

$$\begin{aligned} L_i(\omega^t) - L_i(\omega^{t+1}) &\geq -\beta\eta^t \nabla L_i(\omega^t) \cdot d^t \\ &\geq -\beta\eta^t \max_{j=1,\cdots,m} \nabla L_j(\omega^t) \cdot d^t \\ &\geq \beta\eta^t \|d^t\|^2 \geq \eta_{lb}\|d^t\|^2. \end{aligned} \tag{8}$$

Take the summation from 0 to t, and let $t \to \infty$, then

$$\sum_{t=0}^{\infty} \beta\eta_{lb}\|d^t\|^2 \leq L_i(\omega^0) - L_i(\omega^*) < \infty, \tag{9}$$

which implies

$$\lim_{t\to\infty} \|d^t\| = 0. \tag{10}$$

Hence, S-FedPG can converge. However, when the learning rate η is set too high or s is insufficiently large that the obtained step size in the step size line search procedure does not satisfy the Armijo condition but satisfies the stopping criterion

$$\frac{1}{m}\sum\nolimits_{i=1}^{m} L_i(\omega^t + \eta^t d^t) < \frac{1}{m}\sum\nolimits_{i=1}^{m} L_i(\omega^t). \tag{11}$$

Since d^t is the common descent direction, $\frac{1}{m}\sum_{i=1}^{m} L_i(\omega^t)$ is decreasing and bounded by 0, according to Monotone Convergence Theorem [17], S-FedPG can converge to a local optimum of the following Federated Learning problem

$$\min_{\omega} \frac{1}{m}\sum_{i=1}^{m} L_i(\omega). \tag{12}$$

A.3.2 Convergence analysis of FedPG

In this part, we give the proof of the convergence of FedPG as follows.

Proof:

Assume that all clients are online at each round, and their objectives are differentiable and Lipschitz-smooth (L-smooth) [2]. Since d^t is a common descent direction satisfying $\nabla L(\omega^t)^T d^t < 0, \forall i \in 1, \cdots, m$, by L-smooth and the descent lemma [2] we have

$$\begin{aligned} L_i(\omega^t) - L_i(\omega^t + \eta^t d^t) &\geq -\eta^t \nabla L_i(\omega^t)^T d^t - \frac{1}{2}(\eta^t)^2 \mathbb{L}_i \|d^t\|^2, \\ &= \eta^t(|\nabla L_i(\omega^t)^T d^t| - \frac{1}{2}\eta^t \mathbb{L}_i \|d^t\|^2), \\ &\geq \eta^t(\frac{1}{2}\epsilon|\nabla L_i(\omega^t)^T d^t|), \ \forall i = 1, \cdots, m, \end{aligned} \tag{13}$$

where $\mathbb{L}_i$ represents the Lipschitz constant of each local objective L_i.

Given ω^t and the common descent direction d^t, the cost difference of $L_i(\omega^t) - L_i(\omega^t + \eta^t d^t)$ is majorized by

$$\eta \nabla L_i(\omega^t)^T d^t + \frac{1}{2}\eta^2 \mathbb{L}_i \|d^t\|^2. \tag{14}$$

By the minimum of (14) over η, we can obtain the upper bound of the step size that can reduce the local objective L_i:

$$2 \cdot \frac{|\nabla L_i(\omega^t)^T d^t|}{\mathbb{L}_i \|d^t\|^2}, \tag{15}$$

Thus, the upper bound of the step size that can reduce all local objectives is

$$2 \cdot \min_i \frac{|\nabla L_i(\omega^t)^T d^t|}{\mathbb{L}_i \|d^t\|^2}. \tag{16}$$

Besides, $\eta^t > 0$. Using the step size boundary and the inequality (13), we can obtain the boundary of the improvement of the objectives:

$$L(\omega^t) - L(\omega^t + \eta^t d^t) \geq -\frac{1}{2}\epsilon^2 \nabla L(\omega^t)^T d^t \geq \vec{0}. \tag{17}$$

From the sequence $(\omega^t)_t$, take a subsequence $(\omega^{t_l})_l$. Since $\forall i, L_i(\omega^t)$ is decreasing and bounded by 0, we have

$$\lim_{t\to\infty} ||L(\omega^t) - L(\omega^{t+1})|| = \vec{0}. \tag{18}$$

Hence, by (17), we obtain

$$\lim_{l\to\infty} \nabla L(\omega^{t_l})^T d^{t_l} = \vec{0}, \tag{19}$$

which implies it converges to a Pareto stationary point.

A.3.3 Convergence rate

The convergence of FedPG requires $\|d^t\| \leq \epsilon$. Here ϵ is denoted as the tolerance. In part 1, we will prove that FedPG converges sublinearly. We further prove that if the local objective $L_i(\omega)$ is strongly convex, FedPG converges linearly (see part 2).

[Part 1]

Assume that L_i is differential and L-smooth. Since $d^t = -Q\lambda$ is a direction used to update the model ω at each round t, where $Q = \text{concat}(g_1^t, \cdots, g_m^t, \nabla F(\omega^t))$, and $\lambda \in \mathbb{R}^{m+1}$ is a vector obtained by solving Problem (5) of the main paper.

We rewrite Problem (2) to the following dynamic single objective problem, which is the dynamic weight-sum of the local objectives with the weights vector μ:

$$\min_{\omega^t} \mathcal{F}(\omega^t) = \sum_{i=1}^{m} \mu_i L_i(\omega^t), \tag{20}$$

where $\mu_i = \lambda_i + h_i \lambda_{m+1}$, $h = \frac{1}{\|L(\omega^t)\|} \cdot (\frac{L(\omega^t)^T \vec{1} L(\omega^t)}{\|\vec{1}\| \|L(\omega^t)\|^2} - \frac{\vec{1}}{\|\vec{1}\|})$,

By the definition of $\mathcal{F}(\omega^t)$ we have $\nabla \mathcal{F}(\omega^t) = -d^t$, and F is L-smooth. Define $\mathbb{L}_{\mathcal{F}} := \max_{\mu} \sum_{i=1}^{m} \mu_i \mathbb{L}_i$, we have

$$\begin{aligned} \mathcal{F}(\omega^{t+1}) &\leq \mathcal{F}(\omega^t) + \nabla \mathcal{F}(\omega^t)(\omega^{t+1} - \omega^t) + \frac{\mathbb{L}_{\mathcal{F}}}{2} \|\omega^{t+1} - \omega^t\|^2 \\ &= \mathcal{F}(\omega^t) - \eta^t \|d^t\|^2 + \frac{\mathbb{L}_{\mathcal{F}} \cdot (\eta^t)^2}{2} \|d^t\|^2 \\ &= \mathcal{F}(\omega^t) - [\eta^t - \frac{\mathbb{L}_{\mathcal{F}} \cdot (\eta^t)^2}{2}] \|d^t\|^2. \end{aligned} \tag{21}$$

Consider a fixed $\eta^t = \frac{1}{\mathbb{L}_{\mathcal{F}}}$, which can make $\mathcal{F}(\omega^t) - \mathcal{F}(\omega^{t+1}) \geq 0$ holds. We have:

$$\mathcal{F}(\omega^{t+1}) \leq \mathcal{F}(\omega^t) - \frac{1}{2\mathbb{L}_{\mathcal{F}}} \|d^t\|^2. \tag{22}$$

Take t from 0 to T, and take the summation of them, we have

$$\mathcal{F}(\omega^T) \leq \mathcal{F}(\omega^0) - \frac{1}{2\mathbb{L}_{\mathcal{F}}} \sum_{t=0}^{T-1} \|d^t\|^2. \tag{23}$$

Thus, we can get

$$\frac{1}{2\mathbb{L}_{\mathcal{F}}} \sum_{t=0}^{T-1} \|d^t\|^2 \leq \mathcal{F}(\omega^0) - \mathcal{F}(\omega^T) \leq \mathcal{F}(\omega^0) - \mathcal{F}(\omega^*). \tag{24}$$

Further, we obtain

$$\min_t \|d^t\|^2 \leq \frac{1}{T} \sum_{t=0}^{T-1} \|d^t\|^2 \leq \frac{2\mathbb{L}_{\mathcal{F}} [\mathcal{F}(\omega^0) - \mathcal{F}(\omega^*)]}{T}. \tag{25}$$

Thus,

$$\mathbb{E}(\min_t \|d^t\| \leq \epsilon) = O(\frac{1}{T}), \tag{26}$$

meaning that it needs at least $T = O(\frac{1}{\epsilon})$ rounds to converge, which indicates the sublinear convergence rate of FedPG.

[Part 2]

Assume that the local objective $L_i(\omega)$ are strongly convex with parameter $u > 0$. Denote ω^* is a Pareto stationary solution. According to the strongly convex property, we have

$$L(\omega^*) - f(\omega^t) \geq \nabla L(\omega^t)^T (\omega^* - \omega^t) + \frac{u}{2} \|\omega^* - \omega^t\|^2. \tag{27}$$

Hence,

$$\nabla L(\omega^t)^T(\omega^* - \omega^t) \leq L(\omega^*) - f(\omega^t) - \frac{u}{2}\|\omega^* - \omega^t\|^2. \tag{28}$$

Denote λ as the optimal solution of Problem (3), which satisfies $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$. Since $\nabla L(\omega^t)^T(\omega^* - \omega^t) < \vec{0}$, we can get

$$\lambda^T \nabla L(\omega^t)^T(\omega^* - \omega^t) \leq L(\omega^*) - f(\omega^t) - \frac{u}{2}\|\omega^* - \omega^t\|^2. \tag{29}$$

Since $d^t = -\nabla L(\omega^t)\lambda$, we have

$$(\omega^t - \omega^*) \cdot d^t \leq L(\omega^*) - f(\omega^t) - \frac{u}{2}\|\omega^* - \omega^t\|^2. \tag{30}$$

Besides, since $L_i(\omega)$ is Lipschitz-smooth, by (17), we have

$$\begin{aligned} L(\omega^t) - L(\omega^{t+1}) &\geq -\frac{1}{2}\epsilon^2 \nabla L(\omega^t)^T d^t \\ &\vdots \\ L(\omega^{(\cdot)}) - L(\omega^*) &\geq -\frac{1}{2}\epsilon^2 \nabla L(\omega^{(\cdot)})^T d^t \end{aligned} \tag{31}$$

Take the summation of the left-hand size, and remain only one item of the right-hand size, we have

$$L(\omega^t) - L(\omega^*) \geq \sum_i -\frac{1}{2}\epsilon^2 \nabla L(\omega^i)^T d^t \geq -\frac{1}{2}\epsilon^2 \nabla L(\omega^t)^T d^t. \tag{32}$$

Take $\eta^t \leq \epsilon^2$, we obtain

$$\begin{aligned} L(\omega^t) - L(\omega^*) &\geq -\frac{\eta^t}{2}\nabla L(\omega^t)^T d^t \\ L(\omega^*) - L(\omega^t) &\leq -\frac{\eta^t}{2}\nabla L(\omega^t)^T(-d^t) \\ L(\omega^*) - L(\omega^t) &\leq -\frac{\eta^t}{2}\|d^t\|^2. \end{aligned} \tag{33}$$

Hence, by (30) and (33), we obtain

$$(\omega^t - \omega^*) \cdot d^t \leq -\frac{\eta^t}{2}\|d^t\|^2 - \frac{u}{2}\|\omega^* - \omega^t\|^2. \tag{34}$$

It yields that

$$\begin{aligned} \|\omega^{t+1} - \omega^*\|^2 &= \|\omega^t + \eta^t d^t - \omega^*\|^2 \\ &= \|\omega^t - \omega^*\|^2 + 2\eta^t(\omega^t - \omega^*)^T d^t + (\eta^t)^2\|d^t\|^2 \\ &\leq \|\omega^t - \omega^*\|^2 - u\eta^t\|\omega^t - \omega^*\|^2 - (\eta^t)^2\|d^t\|^2 + (\eta^t)^2\|d^t\|^2 \\ &= (1 - u\eta^t)\|\omega^t - \omega^*\|^2. \end{aligned} \tag{35}$$

Since $(1 - u\eta^t) \in (0, 1)$, we complete the proof that FedPG converges linearly when the local objectives are strongly convex.

In conclusion, FedPG converges sublinearly, and if the local objectives are strongly convex, it converges linearly.

B. Additional Results

In this section, we provide more experimental details and results of FedPG and S-FedPG.

Implementation Details

We adopt CNN [12] on CIFAR-10 [11] by following the previous works [15, 19], which contains 2 convolutional (Conv) layers and 3 fully-connected (FC) layers. The Conv layers have 64 and 64 channels, respectively. The FC layers have 384, 192, and 10 neurons, respectively. Moreover, in Appendix.B.2, we also utilize LeNet-5 [12] on CIFAR-10 as a comparison.

In the main paper, we use Multilayer perceptron (MLP) [16] for FMNIST [20]. We adopt NFResNet-18 [3] on CIFAR-100 [11], which is a more advanced ResNet than the original Resnet for distributed training tasks.

We train every method for at least 2000 rounds. In each round, clients perform local training for 5 epochs, same to [4]. The learning rate is set $\eta \in \{0.01, 0.05, 0.1\}$ with the decay of 0.999 per round, and we choose the best performance of each method in comparison. We take the average of the results in 5 runs with different random seeds.

All experiments are implemented in Pytorch 1.8.1. The Federated Learning is simulated on a server with 64 Intel(R) Xeon(R) Silver 4216 @ 2.10GHz CPUs and 2 NVidia(R) 3090 GPUs.

Baselines

The baselines used in our experiments are listed as follows.

- Local: Local learning without collaboration.
- FedAvg [15]: A traditional federated learning algorithm.
- FedProx [14]: Leveraging control variates via a proximal term to correct local gradients.
- APFL [6]: An FL method to mix the global model and the personalized model.
- pFedMe [18]: An FL method that uses a regularization with Moreau envelope function.
- Ditto [13]: An FL personalized FL method that uses a Multi-task Learning framework in FL.
- FedAMP [10]: An FL method that enhances the collaboration among similar clients.
- FedFomo [22]: An FL strategy that is to collaboratively train a personalized model for each client.
- FedRep [5]: An FL method that learns a local classifier layer and representation layers, respectively.
- FedROD [4]: An FL method that tries to bridge the generic and personalized FL.
- SplitGP [8]: An FL method that split the model into client-side and server-side and take both of them into consideration during the training process.
- FedMGDA+ [9]: Treating FL as a multi-objective optimization.

For FedAvg, FedProx, and FedMGDA+, since they only contain the global model, we follow [4] to use the local training models before the aggregation as their personalized model. Some baselines include extra hyper-parameters, and we use the recommended hyper-parameter setting based on their papers.

Metrics. As mentioned in the main paper, we use L-acc., S-acc., and G-acc. to evaluate the personalization and generalization of the models. Most of the previous personalized federated learning only test L-acc. of the personalized model and show their improvement in L-acc., which cannot evaluate the trade-off between personalization and generalization. Actually, it's useless to increase L-acc. but with a quite low G-acc., because the local learning can easily train a local overfitting model for each client, leading to generating an top L-acc. Besides, S-acc. is considered as a compromise of L-acc. and G-acc., which reflects the performance of the personalized model on the mixed testing data that includes client's local data and the randomly selected test set from other clients. Compared with the metric utilized in SplitGP [8], which also tests the accuracy of the model on a mixed testing dataset, our S-acc. is similar to the case of setting $\rho = (m-1)c$ in SplitGP, where ρ is defined by $\frac{\#\text{others}'\text{ test data}}{\#\text{own test data}}$. But SplitGP only evaluates the accuracy under $\rho \in [0, 1]$, which cannot reveal more about the generalization sacrifice because it only contains a small amount of out-of-client data and ignores the majority of others'. The majority of the mixed testing data is still in the same distribution as clients' own data. Differently, since each client's data may be distributed differently in practice, we build the mixed testing data by "client-wise" combining the testing data: $D_i = D_i \cup D_{k_1} \cup \cdots \cup D_{k_{(m-1)c}}$, where $k_1, \cdots, k_{(m-1)c}$ are clients randomly selected from client j that $j \neq i$. Note that the data amounts of D_i and $D_{k_1}, D_{k_2}, \cdots$ are the same, so that the S-acc. won't be dominated by the data of the specific client and can fairly reflect the accuracy on the mixed test data.

Dataset	FMNIST				CIFAR-10				CIFAR-100			
Model	PM			GM	PM			GM	PM			GM
Accuracy	L-acc.	S-acc.	G-acc.	acc.	L-acc.	S-acc.	G-acc.	acc.	L-acc.	S-acc.	G-acc.	acc.
Local	.989(.018)	.192(.101)	.197(.054)	-	.806(.073)	.166(.096)	.157(.056)	-	.878(.095)	.026(.025)	.014(.013)	-
FedAvg	.989(.019)	.355(.114)	.365(.107)	.411	.925(.046)	.426(.125)	.421(.082)	.597	.921(.065)	.047(.036)	.034(.027)	.191
FedProx	.946(.035)	.762(.086)	.759(.070)	.832	.662(.104)	.548(.078)	.534(.071)	.569	.713(.127)	.164(.049)	.150(.040)	.211
APFL	.992(.014)	.576(.139)	.573(.128)	.849	.925(.048)	.327(.121)	.311(.082)	.631	.918(.066)	.048(.041)	.034(.026)	.176
pFedMe	.946(.034)	.761(.087)	.757(.071)	.830	.437(.187)	.373(.076)	.362(.067)	.384	.690(.137)	.162(.049)	.151(.042)	.203
Ditto	.991(.016)	.378(.124)	.389(.111)	.799	.892(.061)	.189(.106)	.176(.056)	.528	.862(.091)	.025(.026)	.014(.013)	.028
FedAMP	.989(.018)	.192(.101)	.197(.054)	-	.809(.077)	.167(.094)	.159(.056)	-	.878(.095)	.026(.025)	.014(.013)	-
FedFomo	.991(.018)	.192(.101)	.197(.054)	.305	.888(.066)	.184(.103)	.171(.056)	.149	.896(.078)	.026(.025)	.014(.013)	.013
FedRep	.992(.016)	.516(.134)	.513(.120)	.792	.919(.049)	.419(.105)	.414(.091)	.642	.926(.060)	.071(.049)	.062(.036)	.213
FedROD	.991(.016)	.239(.100)	.256(.074)	.448	.909(.053)	.225(.103)	.210(.066)	.498	.845(.106)	.032(.030)	.021(.015)	.059
SplitGP	.975(.046)	.346(.130)	.353(.128)	.573	.806(.070)	.168(.101)	.154(.054)	.223	.770(.124)	.024(.023)	.012(.012)	.053
FedMGDA+	.992(.018)	.364(.128)	.371(.120)	.748	.912(.058)	.214(.110)	.200(.063)	.591	.869(.095)	.026(.025)	.014(.014)	.113
FedPG	.897(.068)	.890(.045)	.890(.047)	.890	.749(.106)	.746(.056)	.742(.062)	.741	.462(.139)	.397(.053)	.402(.060)	.402
S-FedPG	.893(.071)	.891(.047)	.891(.043)	.891,	.747(.105)	.745(.062)	.744(.059)	.741	.455(.139)	.406(.057)	.405(.057)	.401
FedPG γ=0.3	.950(.033)	.860(.056)	.859(.050)	.890	.762(.104)	.746(.053)	.741(.060)	.741	.494(.158)	.384(.060)	.373(.052)	.402
FedPG γ=0.6	.955(.031)	.854(.058)	.853(.052)	.890	.767(.102)	.747(.052)	.741(.063)	.741	.513(.166)	.380(.056)	.369(.056)	.402
FedPG γ=1.0	.956(.031)	.846(.062)	.848(.054)	.890	.770(.101)	.745(.052)	.740(.064)	.741	.521(.165)	.377(.062)	.365(.054)	.402

Table 1. The average (the standard deviation) of L-acc., S-acc., G-acc. of personalized models (PM), and the accuracy of the global model (GM) on three different datasets in Pat-2 with batch size 50, 5 local epochs and 2000 rounds. 10% of 100 clients are online per round.

Proportion	10%				50%				100%			
Model	PM			GM	PM			GM	PM			GM
Accuracy	L-acc.	S-acc.	G-acc.	acc.	L-acc.	S-acc.	G-acc.	acc.	L-acc.	S-acc.	G-acc.	acc.
Local	.806(.073)	.166(.096)	.157(.056)	-	.810(.068)	.164(.094)	.157(.056)	-	.807(.071)	.164(.095)	.157(.058)	-
FedAvg	.925(.046)	.426(.125)	.421(.082)	.597	.918(.055)	.322(.120)	.313(.087)	.625	.919(.053)	.231(.111)	.216(.067)	.624
FedProx	.662(.104)	.548(.078)	.534(.071)	.569	.672(.104)	.552(.073)	.546(.069)	.560	.684(.102)	.544(.076)	.543(.076)	.560
APFL	.925(.048)	.327(.121)	.311(.082)	.631	.923(.053)	.289(.125)	.274(.080)	.636	.923(.050)	.276(.123)	.260(.074)	.638
pFedMe	.437(.187)	.373(.076)	.362(.067)	.384	.450(.187)	.376(.074)	.368(.068)	.370	.453(.184)	.376(.075)	.368(.069)	.387
Ditto	.892(.061)	.189(.106)	.176(.056)	.528	.921(.054)	.243(.118)	.233(.066)	.632	.919(.054)	.265(.119)	.255(.078)	.631
FedAMP	.809(.077)	.167(.094)	.159(.056)	-	.822(.070)	.167(.095)	.154(.058)	-	.808(.077)	.166(.093)	.155(.059)	-
FedFomo	.888(.066)	.184(.103)	.171(.056)	.149	.902(.062)	.190(.106)	.173(.058)	.111	.903(.062)	.188(.104)	.174(.057)	.134
FedRep	.919(.049)	.419(.105)	.414(.091)	.642	.919(.051)	.280(.109)	.263(.072)	.585	.896(.066)	.194(.104)	.181(.059)	.476
FedROD	.909(.053)	.225(.103)	.210(.066)	.498	.910(.053)	.203(.106)	.190(.064)	.459	.883(.060)	.211(.099)	.196(.064)	.409
SplitGP	.806(.070)	.168(.101)	.154(.054)	.223	.890(.062)	.211(.106)	.198(.067)	.456	.923(.054)	.223(.111)	.210(.070)	.609
FedMGDA+	.912(.058)	.214(.110)	.200(.063)	.591	.912(.057)	.213(.106)	.200(.063)	.575	.891(.058)	.196(.104)	.184(.064)	.517
FedPG	.749(.106)	.746(.056)	.742(.062)	.741	.742(.107)	.739(.070)	.739(.062)	.739	.728(.098)	.724(.060)	.725(.063)	.724
S-FedPG	.747(.105)	.740(.062)	.744(.059)	.741	.736(.089)	.727(.068)	.724(.062)	.722	.710(.088)	.709(.065)	.710(.063)	.707
FedPG γ=.3	.762(.104)	.746(.053)	.741(.060)	.741	.786(.088)	.734(.063)	.731(.060)	.739	.808(.076)	.717(.066)	.714(.064)	.724
FedPG γ=.6	.767(.102)	.747(.052)	.741(.063)	.741	.788(.087)	.734(.064)	.731(.061)	.739	.811(.075)	.717(.065)	.714(.065)	.724
FedPG γ=1.0	.770(.101)	.745(.052)	.740(.064)	.741	.790(.085)	.734(.066)	.730(.061)	.739	.813(.074)	.713(.067)	.711(.067)	.724

Table 2. The average (the standard deviation) of L-acc., S-acc., G-acc. of personalized models (PM), and the accuracy of the global model (GM) on CIFAR-10 Pat-2 under 10%, 50%, and 100% of 100 clients online per round, with batch size 50, 5 local epochs and 2000 rounds.

B.1 . Evaluate Personalization & Generalization Under More Scenarios

In Table 1, we evaluate the L-acc., S-acc., and G-acc. of algorithms on FMNIST, CIFAR-10, and CIFAR-100 in Pat-2 with batch size 50 over 2000 communication rounds. It validates that previous PFL approaches suffer from different amounts of global performance reduction in different scenarios and models, revealing that they cannot achieve a robust balance between personalization and generalization.

We further evaluate the performance of P-models and G-models under different proportions of online clients: 10%, 50%, and 100% of 100 clients are randomly selected to be online at each communication round. Table 2 lists the results of the comparison, further justifying the observation of the main paper that previous PFL methods suffer from uncontrollable reductions in generalization performance in different scenarios.

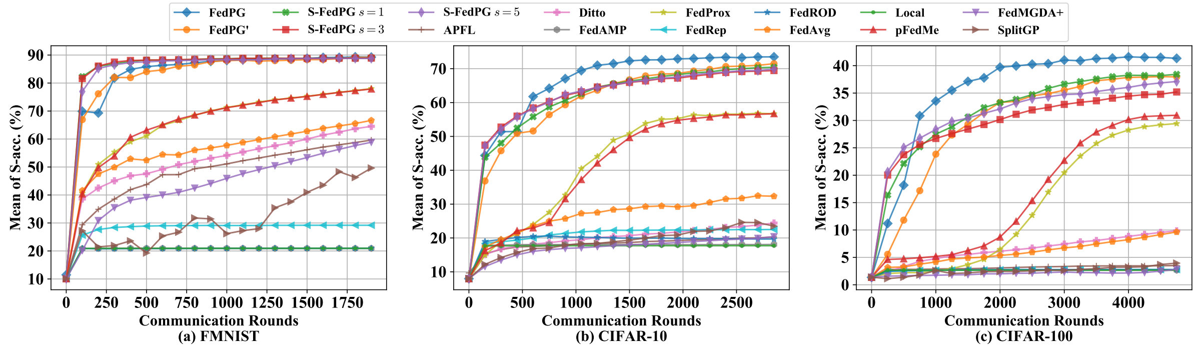

Figure 1. The average S-acc. of clients' P-models in Pat-2 case on (a) FMNIST, (b) CIFAR-10, and (c) CIFAR-100. 100% of 100 clients are online at each communication round.

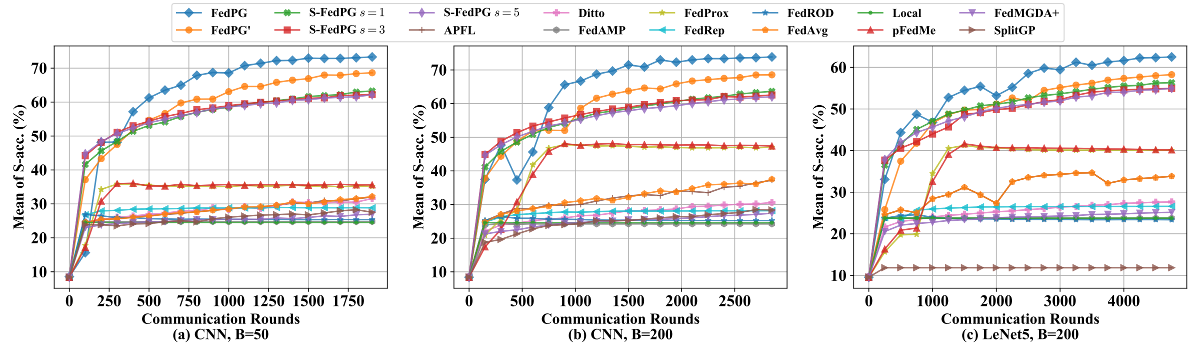

Figure 2. The average S-acc. of clients' P-models in Pat-2 case on CIFAR-10 with 10 clients by three settings: (a) Using CNN and set batch size to 50; (b) Using CNN and set batch size to 200; (c) Using LeNet-5 and set batch size to 200. 100% clients are online at each communication round.

B.2. Convergence and Efficiency

Fig. 1 illustrates the full results of the comparison mentioned in Section 4.2 of the main paper. We observe that the s of S-FedPG is not much sensitive. So we can set a small s, such as $s = 1$ when we use S-FedPG.

In Fig. 2, we further test the convergence efficiency in Pat-2 on CIFAR-10 with only 10 clients on: (a) Using CNN and set batch size to 50; (b) Using CNN and set batch size 200; (c) Using LeNet-5 and set batch size to 200. 100% of 10 clients are online at each communication round. The results verify that FedPG and S-FedPG outperform previous methods, and S-FedPG can stabilize the convergence, but FedPG achieves a higher S-acc. in the later period.

	clean				A1				A2			
Model	PM			GM	PM			GM	PM			GM
Accuracy	L-acc.	S-acc.	G-acc.	acc.	L-acc.	S-acc.	G-acc.	acc.	L-acc.	S-acc.	G-acc.	acc.
FedAvg	.989(.019)	.355(.114)	.365(.107)	.411	.650(.197)	.230(.079)	.239(.067)	.23	.087(.192)	.100(.073)	.105(.045)	.105
FedProx	.946(.035)	.762(.086)	.759(.070)	.832	.683(.202)	.315(.087)	.323(.075)	.333	.759(.233)	.750(.088)	.744(.066)	.744
APFL	.992(.014)	.576(.139)	.573(.128)	.849	.195(.281)	.151(.081)	.131(.060)	.103	.195(.327)	.110(.088)	.118(.050)	.105
pFedMe	.946(.034)	.761(.087)	.757(.071)	.83	.698(.203)	.343(.096)	.345(.074)	.345	.756(.235)	.748(.090)	.744(.064)	.74
Ditto	.991(.016)	.378(.124)	.389(.111)	.799	.940(.040)	.402(.102)	.410(.110)	.513	.931(.059)	.693(.108)	.694(.084)	.696
FedRep	.992(.016)	.516(.134)	.513(.120)	.792	.665(.187)	.230(.077)	.237(.075)	.247	.900(.076)	.171(.096)	.175(.060)	.194
FedROD	.991(.016)	.239(.100)	.256(.074)	.448	.236(.207)	.226(.076)	.229(.058)	.233	.898(.118)	.170(.094)	.176(.065)	.262
SplitGP	.975(.046)	.346(.130)	.353(.128)	.573	.893(.149)	.166(.098)	.169(.058)	.106	.113(.246)	.097(.060)	.093(.038)	.093
FedMGDA+	.992(.018)	.364(.128)	.371(.120)	.748	.989(.021)	.510(.135)	.495(.120)	.705	.921(.091)	.409(.110)	.421(.106)	.591
FedPG	.897(.068)	.890(.045)	.890(.047)	.890	.892(.081)	.882(.048)	.882(.045)	.887	.899(.067)	.891(.050)	.892(.044)	.892

Table 3. The results under no attacks (clean), A1, and A2 attacks, listing the average (the standard deviation) of L-acc., S-acc., G-acc. of personalized models (PM), and the accuracy of the global model (GM) on FMNIST with 100 clients, where 10% are malicious clients.

B.3. Robustness

To verify robustness of the proposed FedPG, we build two common means of attacks to test the robustness of algorithms: (A1) Malicious upload random-zero-mean Gaussian parameters (or gradients for gradient-based methods) to the server [21, 13]; (A2) Malicious clients try to dominate in FL by scaling their updates to be larger, e.g., upload $100(\omega^{t+1} - \omega^t) + \omega^t$ (or $100g_i^t$ for gradient-based FL).

Table. 3 depicts the comparison results on FMNIST with 100 clients where 10% are dishonest clients. We describe that P-models and the G-model of previous methods suffer from various performance reductions under the A1 and A2 attacks. In comparison, FedPG can protect the model's performance of honest clients well under the attacks so that the performances are much more stable.

References

[1] Martin Andersen, Joachim Dahl, and Lieven Vandenberghe. Cvxopt: Convex optimization. *Astrophysics Source Code Library*, pages ascl–2008, 2020. 2

[2] Dimitri P Bertsekas. Nonlinear programming third edition. *Journal of the Operational Research Society*, 48(3):51–51, 2016. 5

[3] Andrew Brock, Soham De, and Samuel L. Smith. Characterizing signal propagation to close the performance gap in unnormalized resnets. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 8

[4] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2021. 8

[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. 8

[6] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020. 8

[7] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical methods of operations research*, 51(3):479–494, 2000. 2

[8] Dong-Jun Han, Do-Yeon Kim, Minseok Choi, Christopher G Brinton, and Jaekyun Moon. Splitgp: Achieving both generalization and personalization in federated learning. *arXiv preprint arXiv:2212.08343*, 2022. 8

[9] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu. Federated learning meets multi-objective optimization. *IEEE Transactions on Network Science and Engineering*, 9(4):2039–2051, 2022. 2, 8

[10] 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 *AAAI*, pages 7865–7873, 2021. 8

[11] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009. 7, 8

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 7

[13] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *ICML*, pages 6357–6368, 2021. 8, 11

[14] 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. 8

[15] 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. 7, 8

[16] Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588, 2009. 8

[17] Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1976. 5

[18] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020. 8

[19] Z. Wang, X. Fan, J. Qi, C. Wen, and R. Yu. Federated learning with fair averaging. In *IJCAI 2021*, 2021. 2, 7

[20] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 8

[21] Xinyi Xu and Lingjuan Lyu. Towards building a robust and fair federated learning system. 2020. 11

[22] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2020. 8