Preserving Global Performance in Personalized Federated Learning

Anonymous ICCV submission

Paper ID 6989

Abstract

Balancing the trade-off between generalization and personalization has been considered a critical problem in Federated Learning (FL). On the one hand, FL with a single global model often has poor personalization that cannot fit every client's data well. On the other hand, Personalized Federated Learning (PFL) methods can fit clients well but are prone to sacrifice model generalization to uncontrollable degrees. To tackle this issue, we propose a Personalized Federated Learning algorithm with Preserved Global Performance (FedPG). We first formulate a multi-objective optimization problem with a fair-driven objective for FL, and calculate a common descent direction to update the global model while simultaneously mitigating the objective bias and absent-client conflicts. A direction-drift method is then well-designed to identify generalization-protected personalized directions to build personalized models. We further show how FedPG can preserve global performance and guarantee convergence. Extensive experiments on different models and learning tasks verify that FedPG outperforms the SOTA PFL algorithms in terms of robustly balancing the trade-off between personalization and generalization.

1. Introduction

Federated Learning (FL) has emerged as a significant machine learning paradigm that can unite clients to collaboratively train models without sharing the private data [22, 47]. It is a popular way to address the problem of data island [50], where each client can train a more generalized model than local learning. The generic FL iterates to update a single global model (G-model) by aggregating the local training results [31]. It has shown great promise when clients' data are IID [31, 55] (i.e., the data and labels are independent and identically distributed). However, in practice, the IID assumption is often not satisfied, and the G-model can easily suffer local performance degradation on some clients, which will lead to poor performance and a lack of personalization [40].

To deal with such statistical heterogeneity, Personalized Federated Learning (PFL) mechanism has received much attention recently [24, 40, 39, 27, 5]. PFL falls between the generic FL and the local learning, aiming to balance generalization and personalization [40, 41]. In PFL, personalization typically requires that the model used by each client fits its local data distribution, demonstrating good accuracy on the local data. And generalization necessitates that, with the assistance of FL training, the employed model can make reliable predictions on the out-of-client data [40, 52, 17]. A model is said to have good global performance if it performs well (e.g., high test accuracy) on all clients' data.

However, it's a huge challenge to balance the trade-off between generalization and personalization in FL. It's not surprising that global performance is easily diminished when enhancing personalization. But the underlying fatal flaw of previous PFLs is that they exhibit unpredictable degrees of global performance degradation in different scenarios (See Sec. 4.1). Some even perform similarly to local learning, where each client's personalized model (P-model) overfits the local data and has very poor generalization. To address this issue, we propose a novel Personalized Federated Learning with Preserved Global Performance (FedPG) that can build P-models for clients while preserving global performance. It contains two steps: **Step 1** is to enhance the global performance of the G-model to better build P-models; **Step 2** is to extend the G-model to build P-model for each client that can enhance personalization while preserving the global performance. Thus, FedPG can achieve a robust balance between generalization and personalization.

In Step 1, there are three challenges to the performance improvement of the G-model:

(A) Gradients conflict, i.e., $\exists i, j, g_i^t \cdot g_j^t < 0$ is a critical issue. Due to the heterogeneity of clients, the direction $\bar{d}^t$ obtained by traditional FL in each communication round t easily conflicts with some clients' gradients, e.g., $\bar{d}^t \cdot g_i^t > 0$, and thus would lead to a decrease in the G-model's performance on these clients [44]. To handle this, Wang et al. [44] design a projection approach, i.e., FedFV, to mitigate gradient conflicts in FL. But when there are more than three clients, it cannot guarantee to achieve a common descent direction that would not cause the performance reduction on

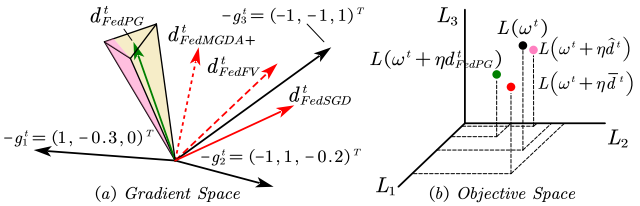

Figure 1. A case of three clients. In (a), the yellow and pink area depict all possible common descent directions d that satisfy $d \cdot g_i^t < 0$, for $i = 1, 2, 3$. The yellow area represents the common descent directions that can further reduce the objective bias. In (b), L_1, L_2 and L_3 depict each client's local objective. $L(\omega^t)$ represents the point in the objective space corresponding to the current model ω^t. $\bar{d}$ describes a direction obtained by previous algorithms. $\hat{d}^t$ in (b) is a direction lying in the pink area in (a).

clients (See Fig. 1 (a)). Besides, Hu et al. [20] propose FedMGDA+, a modified multiple gradient descent algorithm in FL. But it contains a hard-tuned hyper-parameter that cannot ensure the obtained direction is a common descent direction (See Fig. 1 (a)). **(B) Objective bias.** As illustrated in Fig. 1 (b), despite $\hat{d}^t$ is common descent and all local objectives are improved, significant discrepancies exist between the objectives, causing the new model $\omega^t + \eta\hat{d}^t$ to favor client 1 and exhibit poorer performance on client 2 and 3. **(C) Absent-client conflict.** Faced with the challenges of intermittent client availability and communication efficiency [43], only parts of clients can be online in FL at each round. So the direction for the model update may conflict with those absent clients. The model may have very poor performance on them when they return later [44].

To address the above issues, we transfer FL to a multi-objective optimization problem with a fair-driven objective to calculate a direction that is common descent and can reduce the objective bias. We further design an efficient strategy to mitigate the negative impact of client dropout.

In step 2, P-models are extended from the G-model. Some previous PFL approaches [9, 39] utilize a parameter to control the preference of personalization and generalization when building P-models. But they cannot prevent global performance from getting uncontrollable degrees of reduction when increasing personalization. Differently, we design a novel direction-drift method (Sec. 3.3) to compute personalized directions for each client to build their P-model. We further analyze how FedPG converges and can enhance personalization while preserving global performance. We summarize our contributions as follows:

1. We formulate a multi-objective optimization problem with a fair-driven objective for FL and calculate directions that are common descent and can mitigate the objective bias as well as absent-client conflicts.

2. We design a novel direction-drift approach to achieve personalized directions for P-models. We demonstrate that it can enhance personalization while maintaining the global performance of P-models.

3. We introduce an effective way to assess the out-of-client performance of P-models, while most previous PFLs only evaluate the performance on clients' local data that cannot reveal the generalization reduction.

4. We implement extensive experiments on multiple FL scenarios to validate FedPG's superior performance over SOTA methods in terms of balancing the trade-off between personalization and generalization.

2. Background & Related Work

2.1. FL via Multiple Gradient Descent

The generic FL aims to train a global model (G-model) $\omega \in \mathbb{R}^n$ by solving Problem (1) [22, 28]:

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

where L_i represents the local objective of i^{th} client. It is the empirical risk computed from client i's data [30], which is usually defined by a specific loss function such as cross-entropy loss and calculated by $L_i(\omega^t) = \sum_{j=1}^{N_i} \frac{1}{N_i} l_{i_j}(\omega^t)$. N_i represents the number of samples and l_{i_j} is the loss on one sample. However, the generic FL cannot prevent the G-model from suffering the performance reductions on some clients when there exist gradient conflicts, i.e., the local gradient $g_i \cdot g_j < 0$ for some i, j [44]. Differently, there is another way that considers FL as a multi-objective optimization problem (MOP):

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

Definition (Pareto-stationarity [15]): ω^* is called Pareto stationary if and only if there exists a convex combination of the gradients $g_i{=}\nabla L_i(\omega^*)$ that is equal to zero. i.e., $\sum_i^m \xi_i g_i = \vec{0}$, $\xi_i \geq 0$, $\forall i = 1, 2, \cdots, m$, which is equivalent to $\nexists d \in \mathbb{R}^n$, such that $g_i \cdot d < 0$ for all i.

Several gradient-based approaches have been proposed to seek a Pareto stationary solution for Problem (2) [13, 10, 14]. Multiple Gradient Descent Algorithm (MGDA) [13] is a popular method for forcing the objectives of the Problem (2) to be smaller (i.e., $d^t \cdot g_i^t < 0$ for all client i), which solves the MOP by iterating $\omega^{t+1} = \omega^t + \eta^t d^t$, where $d^t \in \mathbb{R}^n$ is an obtained common descent direction. $\eta^t \in \mathbb{R}$ is the learning rate and g_i^t denotes client i's gradient, calculated by $g_i^t{=}\nabla L_i(\omega^t){=}\sum_j^{N_i} \frac{1}{N_i} \nabla L_{i_j}(\omega^t)$. Hu et al. [20] firstly try to use the technique of MGDA in FL and propose FedMGDA+. However, it contains a hyper-parameter that cannot ensure the obtained direction d^t is a common descent direction (See Fig. 1 (a)). We further observe that directly

applying MGDA technique to solve Problem (2) cannot prevent the objective bias across clients in FL, which would harm the global performance of the G-model.

2.2. Personalized Federated Learning (PFL)

Recently, numerous approaches have been utilized to achieve personalization in FL. (1) Using local fine-tuning to accomplish personalization [42, 51]. (2) Employing Multi-task Learning (MTL) [34, 54]: Smith et al. [37] design an MTL framework applicable to convex cases to achieve personalization. (3) Regularizing personalized models towards the global model in a way similar to FedProx [29]: [18, 27, 48]; (4) Enhancing the collaboration among similar clients [21]. (5) Sharing the base of the model and training personalized classifiers for clients [1, 8, 5]. In addition, PFL has also been explored via (6) Model-Agnostic Meta-Learning [12, 39, 38], (7) clustering [7, 35, 36, 3, 16, 11], (8) model sharing [53], (9) Transfer learning [25, 6, 46], etc.

Most of the previous PFL works only show their personalization performance by testing the accuracy of P-models on clients' local data [40], which conceals the issue of generalization degradation. Han et al. [17] evaluate the performance on the mixed data that includes clients' own testing data and a small part of randomly selected global data. But it cannot reveal more about the generalization sacrifice because the majority of the testing data is still in the same distribution as clients' own data. They employ SplitGP [17] to split the model into server & client sides and take both into consideration in the training process. But it still suffers uncontrollable degrees of generalization reduction in different scenarios. Chen & Chao [5] propose FedROD. But clients need to switch the model between the G-model and the P-model if they want to have good generalization or personalization. Conversely, we design a novel direction-drift method to build P-models on the basis of maintaining global performance, allowing us to achieve a robust balance between personalization and generalization.

3. Methodology

The proposed FedPG is shown in Figure 2. Algorithm 1 demonstrates the details of the FedPG's procedure.

3.1. Problem Formulation

To mitigate gradient conflicts and enhance the global performance of G-model in heterogeneous settings, we follow the idea of considering FL as a multi-objective optimization and computing common descent directions for the G-model update (Sec. 2.1). However, we observe that directly using MGDA to solve Problem (2) cannot prevent the objective bias across clients in FL, which would easily make the G-model favor some specific clients while performing poorly on others. As seen in the example of Fig. 1, although the direction $\hat{d}^t$ is common descent, it favors client 1. As a result, the new G-model performs poorly on client 2 and 3. It's undesirable to do personalization based on such a G-model because of its poor global performance. We further show its negative impact in the experiment of Sec. 4.3.

Algorithm 1 FedPG Algorithm

Input: Total communication rounds T, learning rate η.

Initialize G-model parameters ω^0.
for $t = 0, 1, \cdots, T-1$ **do**
  $S^t \leftarrow$ Update the set of online clients.
  Sends ω^t to all client $i \in S^t$ but $i \notin S^{t-1}$.
  for each client i **in parallel do**
    Uploads gradient g_i^t and loss $L_i(\omega^t)$.
  $L^t \leftarrow (L_1(\omega^t), \cdots, L_{|S^t|}(\omega^t))^T$.
  $Q \leftarrow$concat$(g_1^t, \cdots, g_{|S^t|}^t, \nabla F(\omega^t))$.
  Compute λ by solving Problem (5).
  $d^t = -Q\lambda$.
  $d^t = d^t \cdot \| \frac{1}{|S_t|} \sum_i^{|S_t|} g_i^t \| / \|d^t\|$.
  Calculates $\gamma_i, i \in S^t$ based on Section 3.3.
  for each client i **in parallel do**
    Download d^t and γ_i from the server.
    Computes $d_i^t = (-g_i^t - d^t)\gamma_i + d^t$.
    Build P-model $\omega_i^{t+1} = \omega^t + \eta d_i^t$.
  Update $\omega^{t+1} = \omega^t + \eta d^t$ on server and clients.

To address this issue, we design a fair-driven objective $\min F(\omega) = -\frac{L(\omega)\cdot\vec{1}}{\|L(\omega)\|\|\vec{1}\|}$ and add it to Problem (2) to mitigate the objective bias, where $L(\omega)$ denotes a vector of $[L_1(\omega), \cdots, L_m(\omega)]^T$. By maximizing the cosine similarity between $L(\omega)$ and $\vec{1}$, we can drive clients' objectives to be more similar, and thus we can reduce the objective bias. Therefore, the goal of training the G-model is to optimize the following MOP instead of Problem (2).

$$\min_{\omega} \ (L_1(\omega), L_2(\omega), \cdots, L_m(\omega), F(\omega)). \tag{3}$$

Note that the added fair-driven objective doesn't affect ω to converge to a Pareto stationary solution of Problem (2), because when ω is Pareto stationary in Problem (3), it is also Pareto stationary in Problem (2). We give analysis and proof in Appendix.A.1.

When doing personalization, FedPG computes an approximate solution to Problem (4) to obtain a personalized direction for each client to build its P-model.

$$\begin{aligned} &\min_{d_i^t} \ L_i(\omega^t + \eta^t d_i^t), \\ &s.t. \quad g_j^t \cdot d_i^t \leq 0, \forall j \in S^t, j \neq i, \end{aligned} \tag{4}$$

where η^t represents the step size for updating the model at t^{th} communication round, S^t is a set of online clients, and g_j^t denotes other clients' gradients. Hence, client i can build its P-model by $\omega_i^{t+1} = \omega^t + \eta^t d_i^t$. Since the personalized

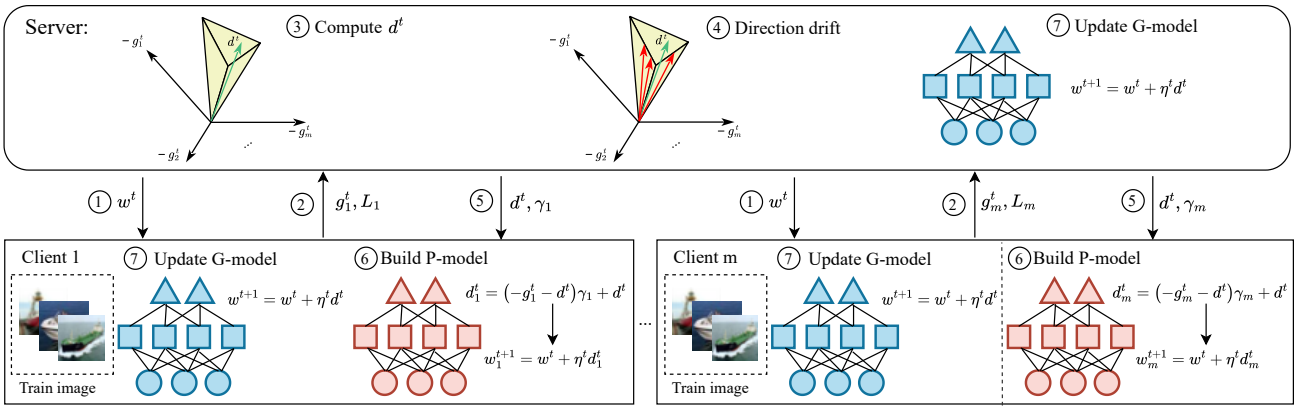

Figure 2. The framework of FedPG, containing seven steps at each communication round t: ① The new-online clients, i.e., clients that were absent before but are online now, download the latest G-model ω^t from the server. ② Client i computes and uploads its local gradient g_i^t and loss $L_i(\omega^t)$ of the G-model. ③ The server calculates a direction d^t for the G-model update and ④ scalars γ_i for personalization. ⑤ Each client i downloads d^t and γ_i. ⑥ Each client i computes its personalized direction d_i^t based on d^t and γ_i and builds personalized model. ⑦ the server and clients update the global model.

direction d_i^t doesn't conflict with others' gradients, it would not lead to decrease the P-model's performance on the out-of-client data and thus preserves the global performance.

3.2. Update the Global Model

3.2.1 Mitigate Gradient Conflicts and Objective Bias

In FedPG, the G-model ω is iteratively updated by $\omega^{t+1} = \omega^t + \eta^t d^t$. To mitigate gradient conflicts in FL, we follow the technique of MGDA to calculate a common descent direction d^t that can drive the objective of each client to be smaller, i.e., $g_i^t \cdot d^t < 0, \forall i$. Differently, since in Sec. 3.1 we add a fair-driven objective $F(\omega)$ in Problem (3) to reduce the objective bias, d^t is also required to satisfy: $\nabla F(\omega^t) \cdot d^t < 0$, which can drive the $F(\omega^t)$ to be smaller.

We first solve the following problem to obtain λ, and then compute d^t by $d^t = -Q\lambda$.

$$\begin{aligned} &\max_{\lambda} -\tfrac{1}{2}\lambda^T (Q^T Q)\lambda \\ &s.t. \ \ \textstyle\sum_{i=1}^{|\lambda|} \lambda_i = 1, \\ &\quad \lambda_i \geq 0, \forall i = 1, 2, \cdots, |\lambda|, \end{aligned} \tag{5}$$

where the matrix $Q \in \mathbb{R}^{n \times (m+1)}$ is the concatenation of the gradients: $Q = \text{concat}(g_1^t, \cdots, g_m^t, \nabla F(\omega^t))$. To prevent from being attacked by some malicious clients who may send fake gradients with a too-small or too-large norm to the server, we drop those zero-norm gradients, and scale $g_1^t, \cdots, g_m^t$ to their average norm before forming the matrix Q. $\nabla F(\omega^t) = \frac{g^t}{\|L(\omega^t)\|} \cdot (\frac{L(\omega^t)^T \vec{1} L(\omega^t)}{\|\vec{1}\| \|L(\omega^t)\|^2} - \frac{\vec{1}}{\|\vec{1}\|})$, where $g^t = \text{concat}(g_1^t, \cdots, g_m^t)$. The obtained d^t satisfies:

1. If ω^t is Pareto stationary, then $d^t = \vec{0}$.
2. If ω^t is not Pareto stationary, then

$$\begin{aligned} &g_i^t \cdot d^t < 0, \ \forall i = 1, \cdots, m, \\ &\nabla F(\omega^t) \cdot d^t < 0. \end{aligned} \tag{6}$$

Therefore, we can obtain a direction for the G-model update that can not only drive the objective of clients to be smaller but also decrease the objective bias. This also indicates that FedPG is robust. Because in FL, the dimension of the model parameters is often in the millions, which is much larger than that of the online clients. Then, one or several malicious clients cannot make the obtained d^t be $\vec{0}$ prematurely by uploading wrong gradients. Thus, even though they upload wrong gradients on purpose, the direction d^t is still a descent direction on other honest clients, which would not cause the performance degradation on honest clients.

3.2.2 Partial Client Participation

Faced with the challenges of intermittent client availability and communication efficiency [43], only parts of clients can participate in FL at t^{th} communication round. So the server can only acquire the local gradients uploaded by the online clients S^t, and thus Problem (3) is decomposed into:

$$\min_{\omega} \ (L_1(\omega), L_2(\omega), \cdots, L_{|S^t|}(\omega), F'(\omega)), \tag{7}$$

where $F'(\omega) = -\frac{L'(\omega) \cdot \vec{1}}{\|L'(\omega)\| \|\vec{1}\|}$ and $L'(\omega)$ is a vector that contains all online clients' loss. At round t, if we set $Q = \text{concat}(g_1^t, g_2^t, \cdots, g_{|S^t|}^t, \nabla F'(\omega))$, and solve Problem (5) to generate a common descent direction d^t for Problem (7), it may not be a descent direction of those absent clients that were online before, which will make the G-model perform worse on these clients if they come back online again. In this regard, we take those absent clients who were online during round $t - \tau$ to $t - 1$ into consideration, where τ represents the expectation of the number of rounds that all the recorded clients participate in FL for one more time. Thus, we have $\tau = M/|S^t|$, where M denotes the number

of recorded clients that have participated in FL so far. In practice, since we cannot directly get the gradients of these clients, we estimate their true gradients according to their recent gradients and combine them with Q in Problem (5) to compute d^t for the G-model update. Since optimizing Problem (5) will lead to a smaller norm of d^t, to prevent the step size from being affected by the norm of d^t, we ultimately scale the length of d^t to be the same as d_r by $d^t{=}\sigma d^t$, where $\sigma = \|d_r\|/\|d^t\|, d^t \neq \vec{0}$. d_r is a direction calculated by the simple aggregation of gradients $g_i, i \in S^t$ as utilized in FedSGD [31], i.e., $d_r = -\sum_i^{|S^t|} \frac{1}{|S^t|} g_i^t$.

Hence, we can mitigate the negative impact of partial client participation/dropout. While the estimated gradients may deviate significantly from the true gradients of those absent clients, it is important to note that the obtained d^t is still a descent direction for online clients and does not lead to a decline in their performance. We show the impact of this strategy in the ablation experiments in Sec. 4.3.

3.3. Build Personalized Models

A direction-drifted procedure is designed to compute a personalized direction for each client to build its P-model.

In Problem (4), we propose a method for building personalized models while preserving the global performance: Computing a personalized direction d_i^t that minimizes $L_i(\omega^t + \eta^t d_i^t)$ while ensuring $d_i^t \cdot g_j^t \leq 0$. However, Problem (4) is an n-dimensional problem, making it computationally challenging in deep learning because n is often on the order of millions. Furthermore, the server needs to send the result d_i^t to each client $i \in S^t$, which incurs a high-communication cost. To address this issue, we design a fast and low-communication-cost approximate method to calculate personalized directions.

Direction drift. Initializing d_i^t to the direction d^t obtained in Sec. 3.2, we drift d_i^t to $-g_i^t$ until it cannot satisfy $g_j^t \cdot d_i^t \leq 0, \forall j \in S^t, j \neq i$. In practice, d_i^t can be calculated by $d_i^t = (-g_i^t - d^t)\gamma_i + d^t$, where $\gamma_i \in [0, 1]$ is the largest value that satisfies inequalities (8).

$$(-g_i^t - d^t)\gamma_i + d^t \leq g_j^t, \forall j \in S^t, j \neq i. \tag{8}$$

Fig. 3 visualizes how we conduct the direction drift to obtain personalized direction d_i^t. Given that $-g_i^t$ represents the steepest descent direction of $L_i(\omega^t)$, d_i^t can be regarded as the steepest common descent direction in the space of Φ_i, spanned by $-g_i^t$ and d^t. Since d_i^t does not interfere with other clients' gradients, it enables improved personalization while preserving generalization performance. Thus, the server only needs to send γ_i to client i, $i \in S^t$, and then the client i can compute its personalized direction by $d_i^t = (-g_i^t - d^t)\gamma_i + d^t$ and further update its P-model by $\omega^t + \eta^t d_i^t$ on its own device. Because $\gamma_i, i \in S^t$ are scalars, the personalization process is low-communication-cost.

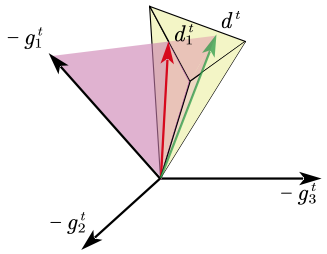

Figure 3. A case of three clients. d^t is a direction for the G-model update. The yellow area depicts all possible common descent directions. The pink area is a part of the space spanned by $-g_1^t$ and d^t. d_1^t lies on the intersection of the yellow area and the pink area.

Since the update of the P-model ω_i^t is independent of its previous state ω_i^{t-1}, clients can opt to construct their P-models only when necessary for inference, conserving computational resources. Moreover, after receiving γ_i from the server, clients can set a larger γ_i to increase the level of personalization in their P-model. This is because a larger γ_i brings d_i^t much closer to $-g_i^t$, prioritizing greater personalization. We discuss the impact of a larger γ in Sec. 4.1.

3.4. Adaptive Step Size via Line Search

Although FedPG employs a common descent direction d^t, the use of a fixed or simple-decayed learning rate η^t at round t may sometimes exceed the step size bound (10), which would decrease the model performance on some clients and lead to convergence oscillation. Therefore, in this section, we propose a low-communication-cost approach to search for a suitable step size η^t.

Starting from $\eta^t{=}2^s\eta$, where s is a given integer that controls the range of step size searching, we repeatedly update $\eta^t{=}\eta^t/2$ until η^t is smaller than its lower bound $(1/2)^s\eta\sigma$ or satisfies the following Armijo condition [32]:

$$\begin{aligned} &L_i(\omega^t) \geq L_i(\omega^t + \eta^t d^t) - \beta\eta^t g^t d^t, \forall i \in S^t \\ &F(\omega^t) \geq F(\omega^t + \eta^t d^t) - \beta\eta^t \nabla F(\omega^t) \cdot d^t, \end{aligned} \tag{9}$$

where g^t is a matrix that concludes gradients uploaded by clients, and $\beta \in (0, 1)$ is a parameter in Armijo condition. Every time η^t is updated, it will be sent to all online clients. Afterwards, each client builds a temporary model and delivers its loss to the server. Then the server determines if η^t satisfies the stopping criterion.

The step size line search is embedded in FedPG to replace the fixed or simple decayed learning rate. We name this variant as S-FedPG and provide more details and the pseudocode in Appendix.A.2. We recommend using the step size line search when the clients can rapidly compute the model's loss, as it incurs low-communication-cost in such scenarios. Firstly, at each round of the step size line search, data transmission between the server and clients only consists of scalars, which costs a tiny amount of bandwidth. Besides, in step size searching, clients only need to

compute the loss through forward propagation, which isn't time-consuming when using the techniques of parallel computation [26]. In some advanced devices such as the memristor [49], the time complexity of the matrix multiplication, which is the fundamental operation of the forward pass, can even reach O(1). Thus, it doesn't take too much time to wait for clients to compute the model loss and reply.

3.5. Convergence Analysis

We analyze the convergence of FedPG and S-FedPG. Assume that clients' objectives are Lipschitz-smooth [2] with constant $\mathbb{L}_i$, all clients are online at each round, and the step size η^t is in the following step size bound,

$$0 < \eta^t \leq 2 \cdot \min_i \frac{|\nabla L_i(\omega^t) \cdot d^t|}{\mathbb{L}_i \|d^t\|^2}, \tag{10}$$

then FedPG can converge to a Pareto stationary solution in a sublinear convergence rate. For S-FedPG, since it utilizes step size line search to obtain a proper step size, the assumption of Lipschitz-smooth can be relaxed to smooth. The detailed proof can be seen in Appendix.A.3.

4. Evaluation

Metrics. Most of the previous PFL studies only test the mean accuracy of P-models on clients' local test sets, which only represent personalized performance and will conceal the generalization reduction of the model. In practice, the test sets are subject to change, necessitating models with good local and out-of-client performance. Otherwise, if the generalization ability is as poor as the local learning, clients cannot benefit from FL and will lose the willingness to join in. Hence, we utilize the following three accuracy metrics to evaluate the performance of PFL in image classification.

- Local accuracy (L-acc.) - Accuracy of P-models on clients' local test sets, which can evaluate the pure personalized performance but totally conceal the generalization ability of the models.
- Generalization accuracy (G-acc.) - Accuracy of the G-model/P-models on test sets of all clients, which represents the global performance of the model.
- Synthetic accuracy (S-acc.) - Denote $D_1, \cdots, D_m$ as the test data of each client with the same data amount. For each client i, we evaluate S-acc. on the mixed testing data built by "client-wise" combining D_i and $D_{k_1}, \cdots, D_{k_{(m-1)c}}$, where $k_1, \cdots, k_{(m-1)c}$ are $c\%$ of clients other than i that are randomly selected. c is defaulted set to 50%. S-acc. is considered a compromise of L-acc. and G-acc., which reflects the P-model's performance on the local data and parts of the out-of-client data.

Datasets and Models. We use CIFAR-10/100 [23] and Fashion MNIST (FMNIST) [45], where the local training and test dataset are already split. We consider two non-IID scenarios. (1) Dir(α): Follow [19, 5] to create a Dirichlet heterogeneous partition for m clients. With α<1, most of the training/test data of one class is probably assigned to a small portion of clients. (2) Pat-2: We follow [31, 21] to make a pathological non-IID data setting that assigns each client to two classes of samples with different amounts. At each communication round, p% of 100 clients are randomly selected to join in FL. Particularly, We use CNN [31] for CIFAR-10, adopt Multilayer perceptron (MLP) [33] for FMNIST, and utilize NFResNet-18 [4] for CIFAR-100.

Baselines and Hyper-parameters. We compare with the traditional FL method, FedAvg [31], the local learning (Local), FedProx [29], FedMGDA+ [20], personalized FL algorithms, including pFedMe [39], Ditto [27], FedAMP [21], FedFomo [53], FedRep [8], FedROD [5], SplitGP [17]. We directly rewrite the code of the compared algorithms based on the authors' open-source code. Because FedAvg, FedProx, and FedMGDA+ only contain global models, we follow [5] to adopt their local training results before the aggregation as their P-models to test performance. For Local and FedAMP, there are no G-models. All clients use Stochastic Gradient Descent (SGD) on local datasets with 5 local epochs and the learning rate $\eta \in \{0.01, 0.05, 0.1\}$ decay of 0.999 per round, and we choose the best performance of each method in comparison. We take the average results in 5 runs with different random seeds.

4.1. Evaluate Personalization & Generalization

We first evaluate the average L-acc., S-acc., and G-acc. of P-models (denoted as PM in the table) to test the personalized and generalized performance of the proposed FedPG and S-FedPG in Dir(0.1) on FMNIST, CIFAR-10, and CIFAR-100 over 2000 rounds. 10% of 100 clients are randomly selected at each round. Meanwhile, we calculate the G-acc. of the G-model (denoted as GM in the table) as a comparison to examine the reduction of the global performance of P-models.

Table 1 lists the results. In terms of the mean S-acc. and G-acc., FedPG and S-FedPG markedly outperform the previous representative algorithms. Compared with the accuracy of the G-model, P-models of FedPG maintain the global performance and achieve an increase in L-acc. This is because FedPG adopts a generalization-protected direction to build personalized models. Moreover, we evaluate the case that clients using FedPG adopt a larger γ to further enhance personalization, i.e., FedPG γ=0.3, FedPG γ=0.6, and FedPG γ=1.0. The results depict that a larger γ can further increase L-acc., but sacrifice S-acc. and G-acc. a little (but they are still much higher than that of the previous). In Appendix.B.1, we present more results in different scenar-

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	.947(.063)	.300(.129)	.295(.124)	-	.788(.189)	.175(.095)	.150(.077)	-	.878(.095)	.026(.025)	.014(.013)	-
FedAvg	.974(.036)	.760(.091)	.761(.095)	.876	.880(.150)	.461(.166)	.431(.139)	.571	.597(.087)	.255(.045)	.251(.043)	.326
FedProx	.939(.054)	.763(.102)	.779(.091)	.838	.666(.154)	.556(.125)	.527(.121)	.556	.713(.127)	.164(.049)	.150(.040)	.211
APFL	.975(.036)	.758(.082)	.759(.092)	.866	.793(.158)	.605(.122)	.566(.138)	.643	.600(.087)	.250(.045)	.253(.045)	.322
pFedMe	.938(.057)	.762(.100)	.773(.094)	.810	.683(.147)	.550(.121)	.526(.129)	.560	.690(.137)	.162(.049)	.151(.042)	.203
Ditto	.964(.044)	.650(.105)	.659(.106)	.842	.860(.135)	.255(.168)	.217(.109)	.526	.862(.091)	.025(.026)	.014(.013)	.028
FedAMP	.947(.063)	.300(.129)	.295(.124)	-	.794(.187)	.171(.096)	.147(.076)	-	.423(.108)	.055(.027)	.051(.019)	-
FedFomo	.937(.099)	.313(.127)	.295(.098)	.508	.820(.165)	.178(.097)	.153(.080)	.242	.436(.096)	.058(.029)	.053(.019)	.031
FedRep	.968(.048)	.720(.097)	.725(.090)	.841	.877(.121)	.553(.141)	.516(.145)	.645	.580(.089)	.305(.043)	.306(.043)	.340
FedROD	.966(.045)	.736(.096)	.744(.109)	.862	.845(.168)	.369(.168)	.341(.154)	.601	.447(.075)	.307(.051)	.302(.046)	.312
SplitGP	.894(.156),	.457(.108)	.469(.120)	.710	.789(.188)	.166(.107)	.153(.084)	.267	.341(.119)	.041(.026)	.034(.020)	.104
FedMGDA+	.961(.053)	.621(.116)	.628(.120)	.832	.859(.136)	.321(.184)	.282(.165)	.569	.559(.089)	.114(.036)	.111(.033)	.269
FedPG	.889(.077)	**.879(.064)**	.895(.051)	.885	.767(.113)	.742(.090)	**.739(.129)**	**.739**	.451(.071)	.426(.051)	.422(.049)	.419
S-FedPG	.887(.078)	.883(.087)	**.897(.050)**	**.885**	.763(.113)	**.760(.090)**	.738(.130)	.737	.461(.071)	**.437(.052)**	**.428(.049)**	**.428**
FedPG γ=0.3	.949(.044)	.852(.071)	.866(.060)	.885	.784(.109)	.762(.113)	.735(.143)	.739	.451(.076)	.408(.050)	.405(.046)	.419
FedPG γ=0.6	.952(.045)	.841(.071)	.849(.068)	.885	.788(.110)	.760(.111)	.734(.143)	.739	.463(.077)	.411(.053)	.402(.046)	.419
FedPG γ=1.0	.953(.042)	.839(.070)	.848(.072)	.885	.789(.107)	.759(.109)	.733(.142)	.739	.467(.077)	.409(.052)	.399(.046)	.419

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 Dir(0.1) with batch size 50, 5 local epochs and 2000 rounds. 10% of 100 clients are online per round.

ios such as Pat-2, and set different client online proportions such as 50% and 100% to make more comparisons.

The results above also verify that protecting global performance while performing personalization can lead to a solid balance between generalization and personalization across different scenarios. In contrast, previous PFL algorithms suffer different degrees of reductions on S-acc. and G-acc. in various cases. For instance, on FMNIST/CIFAR-10/CIFAR-100, Dir(0.1), G-acc. of APFL's P-models shows about 12.3%/12%/21.4% reductions compared to the G-model, while that of pFedMe has 4.6%/6%/21.4% decreases. These unstable reductions also happen in other previous PFL methods. Some PFL models, such as FedAMP and FedFomo (on CIFAR-10), and Ditto (on CIFAR-100), perform nearly the same as the local learning (Local). Their P-models overfit clients' local training samples and lose the generalization ability benefited from FL.

Moreover, we evaluate S-acc. under different c: $c = 0\%$, 1%, 10%, 30%, 50%, 70%, and 100%. When c=0%, S-acc. is the same as L-acc., while G-acc. is equivalent to S-acc. if c=100%. Fig. 4 depicts the comparisons of different algorithms, showing that FedPG achieves more stable and better test accuracy on the mixed test set of different proportions of other clients, while previous PFL approaches suffer a significant accuracy reduction when $c > 0\%$.

4.2. Convergence and Efficiency

We compare the convergent efficiency of FedPG, S-FedPG, and several previous PFL algorithms in the Pat-2 scenario on FMNIST, CIFAR-10, and CIFAR-100 with batch size 50. All 100 clients are online at each round. We

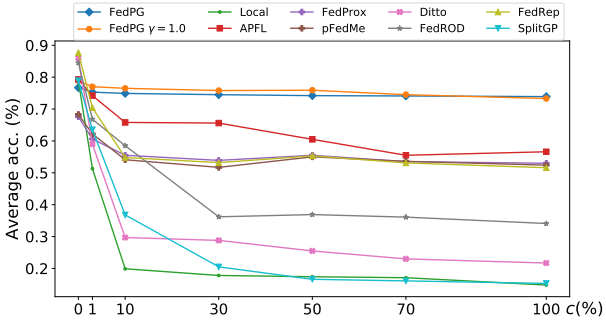

Figure 4. The mean S-acc. under different c in the case of Dir(0.1) on CIFAR-10 with batch size 50, 5 local epochs and 2000 rounds. 10% of 100 clients are online per round.

use S-acc. to evaluate the performance of P-model on the local and out-of-client data. Fig. 5 shows part of the results of compared algorithms, where approaches with comparable performance to others are omitted from the visualization. Full results and more comparisons can be seen in Appendix.B.2.

We unveil that FedPG and S-FedPG can converge much faster and reach a much higher average S-acc. in comparison to prior PFL methods. We attribute this to the more effective direction employed for the G-model update and the use of a generalization-protected direction for building P-models.

Compared with FedPG, S-FedPG converges faster in the early stage and is more stable during the whole training procedure. This is due to the fact that, although the direction d^t can drive reduce each of the objectives of Problem (3), the learning rate we utilized in FedPG in this experiment is

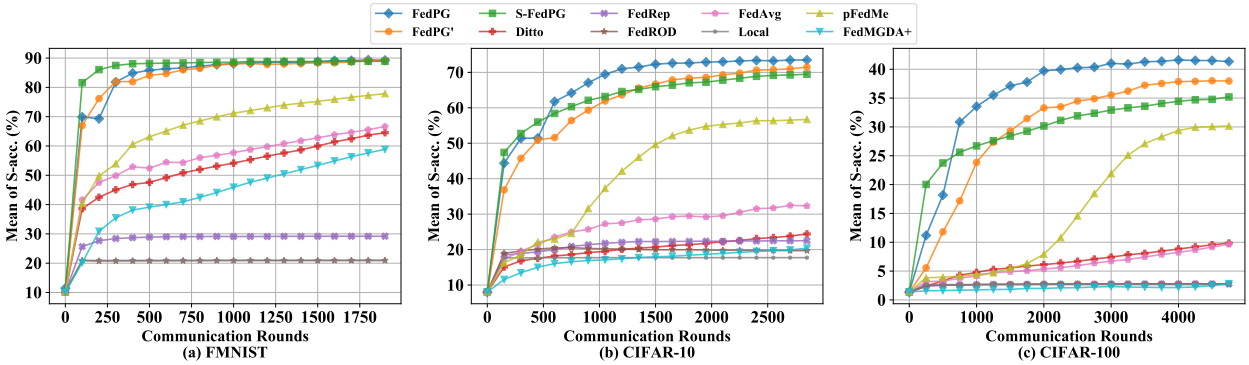

Figure 5. The average S-acc. of clients' P-models in Pat-2 case on (a) FMNIST, (b) CIFAR-10, and (c) CIFAR-100. 100% of 100 clients are online at each communication round.

a simple manual value with decay 0.999 per round, which may exceed outside the step size bound (10) and thus cause the convergence concussion (See Fig. 5.(a)). If we tune a lower learning rate η, (see FedPG' in Fig. 5), it can stabilize the convergence but with a slower rate. For S-FedPG, since it uses a step size line search to obtain a more proper step size to update models, it converges more stably. However, since the model is closer to the convergence in the middle and late stages, it is easier to obtain a step size η^t that is considerably smaller than η used in FedPG. Overall, S-FedPG converges slower than FedPG in the later period.

4.3. Ablation Experiments

We design several modified algorithms to study the effect of each part of FedPG. (**M1**) Replace the common descent direction with d_r by simple aggregation (similar to FedSGD [31]), which cannot ensure to reduce each client' objective. (**M2**) Remove the added fair-driven objective $F(\omega)$ in Problem (3). (**M3**) Replace the personalized direction with the direction used in G-model update, meaning that the P-models are the same as the G-model. (**M4**) Never use the strategy of mitigating the absent-client conflict mentioned in Sec. 3.2.2.

Results of the comparison are presented in Table. 2, demonstrating that without the direction that can drive to improve clients' objectives at each round, the accuracy of the G-model of M1 is decreased to 56.3% on CIFAR-10, which is much lower than that of FedPG (74.1%). Besides, without the added fair-driven objective $F(\omega)$ in Problem (3), which can reduce the objective bias during the FL training, the performance of **M2** is hampered and the standard deviation values are higher. It reflects that there is a higher level of performance bias among clients' objectives, which has a negative impact on the model performance. For **M3**, as it uses the same direction of the G-model update when building the P-model, the performance is poor in personalization. For **M4**, since only part of clients are randomly selected to be online at each round, the direction for G-model and P-models update may conflict with the gradients of those absent clients, and thus the performance of models get much poorer.

	FMNIST				CIFAR-10			
	PM			GM	PM			GM
	L-acc.	S-acc.	G-acc.	acc.	L-acc.	S-acc.	G-acc.	acc.
FedPG	**.897** (.068)	**.890** (.045)	**.890** (.047)	**.890** (.068)	**.749** (.106)	**.746** (.056)	**.742** (.062)	**.741** (.060)
M1	.820 (.152)	.784 (.069)	.777 (.062)	.778 (.061)	.628 (.126)	.565 (.070)	.566 (.070)	.563 (.070)
M2	.859 (.071)	.850 (.053)	.843 (.055)	.841 (.057)	.635 (.113)	.629 (.073)	.629 (.078)	.629 (.078)
M3	.895 (.064)	.889 (.064)	.890 (.068)	.890 (.068)	.749 (.110)	.745 (.058)	.741 (.060)	.741 (.060)
M4	.867 (.089)	.841 (.056)	.836 (.057)	.844 (.056)	.698 (.091)	.637 (.070)	.643 (.070)	.651 (.064)

Table 2. The mean (and the standard deviation) accuracy on FMNIST and CIFAR-10 of Pat-2 in ablation experiments. 10% of 100% clients are online at each round.

5. Conclusion and Future Work

Most existing PFL methods suffer different degrees of generalization reduction in various scenarios, and thus are not resilient. We propose a promising framework to achieve a robust balance between personalization and generalization in FL. Concretely, we first formulate a multi-objective optimization problem with a fair-driven objective for FL and update the G-model by computing a direction that is common descent and can mitigate the objective bias and absent-client conflicts. Then we design a direction-drift approach to build P-models without sacrificing global performance. Extensive experiments on different models and learning tasks demonstrate that FedPG can achieve a robust balance between generalization and personalization in FL. There are a number of intriguing future research topics, such as making more accurate predictions about the gradients of absent clients for better handling the problem of clients' dropout.

References

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

[2] Dimitri P Bertsekas. Nonlinear programming third edition. *Journal of the Operational Research Society*, 48(3):51–51, 1997. 6

[3] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020. 3

[4] 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. 6

[5] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2021. 1, 3, 6

[6] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93, 2020. 3

[7] Yae Jee Cho, Jianyu Wang, Tarun Chiruvolu, and Gauri Joshi. Personalized federated learning for heterogeneous clients with clustered knowledge transfer. *arXiv preprint arXiv:2109.08119*, 2021. 3

[8] 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. 3, 6

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

[10] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012. 2

[11] Moming Duan, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan. Fedgroup: Efficient federated learning via decomposed similarity-based clustering. In *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 228–237. IEEE, 2021. 3

[12] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020. 3

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

[14] Jorg Fliege and A Ismael F Vaz. A method for constrained multiobjective optimization based on sqp techniques. *SIAM Journal on Optimization*, 26(4):2091–2119, 2016. 2

[15] Bennet Gebken, Sebastian Peitz, and Michael Dellnitz. On the hierarchical structure of pareto critical sets. *Journal of Global Optimization*, 73(4):891–913, 2019. 2

[16] 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. 3

[17] 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. 1, 3, 6

[18] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020. 3

[19] 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. 6

[20] 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, 6

[21] 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*, pages 7865–7873, 2021. 3, 6

[22] 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. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. 1, 2

[23] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009. 6

[24] 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

[25] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019. 3

[26] Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. Pytorch distributed: Experiences on accelerating data parallel training. *Proc. VLDB Endow.*, 13(12):3005–3018, 2020. 6

[27] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368, 2021. 1, 3, 6

[28] 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. 2

[29] 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. 3, 6

[30] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 2

[31] 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, 5, 6, 8

[32] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999. 5

[33] 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. 6

[34] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 3

[35] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3710–3722, 2020. 3

[36] Nir Shlezinger, Stefano Rini, and Yonina C Eldar. The communication-aware clustered federated learning problem. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2610–2615. IEEE, 2020. 3

[37] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in Neural Information Processing Systems*, 30, 2017. 3

[38] Jaehun Song, Min-hwan Oh, and Hyung-Sin Kim. Personalized federated learning with server-side information. *arXiv preprint arXiv:2205.11044*, 2022. 3

[39] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020. 1, 2, 3, 6

[40] 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

[41] Xueyang Tang, Song Guo, and Jingcai Guo. Personalized federated learning with contextualized generalization. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2241–2247, 2022. 1

[42] 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. 3

[43] Shiqiang Wang and Mingyue Ji. A unified analysis of federated learning with arbitrary client participation. *arXiv preprint arXiv:2205.13648*, 2022. 2, 4

[44] Z. Wang, X. Fan, J. Qi, C. Wen, and R. Yu. Federated learning with fair averaging. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021. 1, 2

[45] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[46] Hongwei Yang, Hui He, Weizhe Zhang, and Xiaochun Cao. Fedsteg: A federated transfer learning framework for secure image steganalysis. *IEEE Transactions on Network Science and Engineering*, 8(2):1084–1094, 2020. 3

[47] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019. 1

[48] Ruihong Yang, Junchao Tian, and Yu Zhang. Regularized mutual learning for personalized federated learning. In *Asian Conference on Machine Learning*, pages 1521–1536. PMLR, 2021. 3

[49] Peng Yao, Huaqiang Wu, Bin Gao, Jianshi Tang, Qingtian Zhang, Wenqiang Zhang, J. Joshua Yang, and He Qian. Fully hardware-implemented memristor convolutional neural network. *Nature*, 577(7792):641–646, Jan. 2020. 6

[50] Bin Yu, Wenjie Mao, Yihan Lv, Chen Zhang, and Yu Xie. A survey on federated learning in data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(1):e1443, 2022. 1

[51] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020. 3

[52] Honglin Yuan, Warren Morningstar, Lin Ning, and Karan Singhal. What do we mean by generalization in federated learning? *arXiv preprint arXiv:2110.14216*, 2021. 1

[53] 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. 3, 6

[54] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 3

[55] Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3219–3227. International Joint Conferences on Artificial Intelligence Organization, 7 2018. 1