Flexible Sharpness-Aware Personalized Federated Learning

Xinda Xing[1, 2*], **Qiugang Zhan**[3, 4, 5*], **Xiurui Xie**[1†], **Yuning Yang**[2], **Qiang Wang**[6], **Guisong Liu**[3, 4, 5†]

[1]Laboratory of Intelligent Collaborative Computing, University of Electronic Science and Technology of China, Chengdu, China
[2]Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[3]Complex Laboratory of New Finance and Economics, Southwest University of Finance and Economics, Chengdu, China
[4]Engineering Research Center of Intelligent Finance, Ministry of Education, Chengdu, China
[5]Kash Institute of Electronics and Information Industry, China
[6]School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University, Shenzhen, China
{xingxd, yangyuning}@std.uestc.edu.cn, xiexiurui@uestc.edu.cn, {zhanqg, gliu}@swufe.edu.cn, wangq637@mail2.sysu.edu.cn

Abstract

Personalized federated learning (PFL) is a new paradigm to address the statistical heterogeneity problem in federated learning. Most existing PFL methods focus on leveraging global and local information such as model interpolation or parameter decoupling. However, these methods often overlook the generalization potential during local client learning. From a local optimization perspective, we propose a simple and general PFL method, **Fed**erated learning with **F**lexible **S**harpness-**A**ware Minimization (**FedFSA**). Specifically, we emphasize the importance of applying a larger perturbation to critical layers of the local model when using the Sharpness-Aware Minimization (SAM) optimizer. Then, we design a metric, perturbation sensitivity, to estimate the layer-wise sharpness of each local model. Based on this metric, FedFSA can flexibly select the layers with the highest sharpness to employ larger perturbation. Extensive experiments are conducted on four datasets with two types of statistical heterogeneity for image classification. The results show that FedFSA outperforms seven state-of-the-art baselines by up to 8.26% in test accuracy. Besides, FedFSA can be applied to different model architectures and easily integrated into other federated learning methods, achieving a 4.45% improvement.

Code and Appendix — https://github.com/xxdznl/FedFSA

1 Introduction

Traditional federated learning (FL) methods (McMahan et al. 2017; Li et al. 2020; Karimireddy et al. 2020) focus on obtaining one effective global model through collaborative learning. However, in practice, a single global model often fails to meet the needs of all clients due to the typically non-identically and independently distributed (non-IID) nature of client data (Kairouz et al. 2021; Tan et al. 2022; Ye et al. 2023). Personalized federated learning (PFL) effectively alleviates the non-IID problem by customizing personalized models for each client. Previous research on PFL has primarily based on model interpolation (Hanzely and Richtárik 2020; Ma et al. 2022; Zhang et al. 2023b; Wu et al. 2023), as well as parameter decoupling (Arivazhagan et al. 2019; Collins et al. 2021; Xu, Tong, and Huang 2023; Zhang et al. 2023a).

However, existing PFL methods mainly focus on effectively leveraging global and local information, neglecting the inherent generalization potential during local training. Therefore, recently, there has been a new trend to improve the local learning process of clients by applying optimizer methods from centralized learning to FL (Reddi et al. 2020; Caldarola, Caputo, and Ciccone 2022; Qu et al. 2022; Zhou and Li 2023). Among these methods, the Sharpness-Aware Minimization (SAM) optimizer (Foret et al. 2021) used in FedSAM (Caldarola, Caputo, and Ciccone 2022) and MoFedSAM (Qu et al. 2022) has shown outstanding generalization ability in centralized learning. SAM solves a min-max problem by minimizing both the training loss and sharpness, aiming to achieve a flat loss landscape and thereby improve the model's generalization performance. SAM introduces a perturbation when addressing the maximization problem.

Recent research on SAM has shown promising findings on this perturbation. SSAM (Mi et al. 2022) and SAMON (Mueller et al. 2024) suggest that perturbing all parameters may not be necessary. Additionally, DISAM (Zhang et al. 2024) has demonstrated, through both theoretical analysis and experiments, that the larger perturbation can enhance the model's generalization ability, provided that convergence is ensured. For the FL scenarios, PLGU (Qu et al. 2023) and FedSOL (Lee et al. 2024) investigate the application of perturbation to specific layer parameters. Additionally, they have devised an adaptive method for determining the perturbation amplitude based on either the level of model personalization or the disparity between local and global models. Though these methods enhance the model's generalization performance, they barely investigate the influence of perturbation amplitude and the perturbed parameters from the

*These authors contributed equally.
†Corresponding authors.

perspective of sharpness.

These recent works inspire us to rethink how to select the parameters to perturb and the appropriate perturbation amplitude when applying SAM in FL. Therefore, we propose a novel PFL method, called **Fed**erated Learning with **F**lexible **S**harpness-**A**ware Minimization (**FedFSA**). Specifically, we design the perturbation sensitivity metric to estimate sharpness. Based on this metric, FedFSA customizes the SAM optimizer for each client by selecting the layers with the highest sharpness relative to the global model and applying larger perturbations to these layers, aiming for higher generalization performance. Additionally, FedFSA can be seamlessly integrated with momentum to further assist clients in escaping local flat minima. Our contributions can be summarized as:

- We analyze the sharpness in relationship to perturbation and then design a novel metric, perturbation sensitivity, by using perturbation to approximate layer-wise sharpness. To leverage the generalization-enhancing properties of larger perturbation in SAM for personalized federated learning, we further design the global perturbation sensitivity metric.
- Guided by global perturbation sensitivity, which evaluates critical layers based on the sharpness between the local and global models, We propose FedFSA, a flexible sharpness-aware personalized federated learning method, which flexibly applies larger perturbation to critical layers for different clients, thereby enhancing clients' generalization.
- We conducted extensive experiments on Fashion-MNIST, CIFAR10, CIFAR100, and Tiny-ImageNet datasets with two types of non-IID data partitions. Compared to the state-of-the-art methods, FedFSA achieves a significant performance improvement of 8.26% while maintaining computational and communication overheads similar to MoFedSAM. Additionally, FedFSA demonstrated the applicability to different model architectures and various federated learning methods with up to 4.45% improvement.

2 Related Works

This section first reviews research on perturbation in SAM within centralized learning and then discusses the application of SAM in FL scenarios.

2.1 Research on SAM perturbation

Smooth loss landscape is generally associated with better generalization capabilities (Keskar et al. 2016; Neyshabur et al. 2017; Li et al. 2018; Izmailov et al. 2018). Various studies have explored SAM and its perturbation.

Studies on perturbation amplitude (Liu et al. 2022; Ahn, Jadbabaie, and Sra 2024) show that random perturbation can help escape the sharp minima. (Si and Yun 2024) proves that a constant perturbation may lead SAM to converge to additive factors proportional to the square of the perturbation amplitude. To address the limitations of constant perturbation, DSAM (Chen, Li, and Chen 2024) dynamically adjusts the perturbation neighborhood of SAM based on local loss surface properties. DISAM (Zhang et al. 2024) addresses the issue of inconsistent SAM convergence across domains by minimizing variance in domain loss during sharpness estimation. DISAM prevents excessive or insufficient perturbation and demonstrates that larger perturbations can enhance generalization, albeit potentially at the expense of convergence speed.

Studies on perturbation scope Neural network models commonly possess a hierarchical nature. (Zhang, Bengio, and Singer 2022) highlights the importance of considering the unique contributions of each layer in the model, rather than treating the network as a monolithic block. LLMC (Adilova et al. 2024) further supports this by showing that models exhibit hierarchical robustness to perturbation. As for SAM, SSAM (Mi et al. 2022) reveals that perturbing only a small subset of parameters can maintain or even enhance SAM performance. (Lyu, Li, and Arora 2022) suggests that normalization layers can reduce the sharpness of the loss surface, contributing to a flatter loss landscape. Furthermore, SAMON (Mueller et al. 2024) experimentally verifies that applying SAM perturbation solely to batch normalization layers can achieve or exceed the effects of perturbing the entire model.

2.2 Federated SAM-based approaches

FedSAM (Caldarola, Caputo, and Ciccone 2022) is the pioneer in applying SAM to FL. Based on this, MoFedSAM (Qu et al. 2022) introduces an enhancement by incorporating local momentum. PLGU (Qu et al. 2023) adaptively applies perturbations to layers based on their degree of personalization. FedSOL (Lee et al. 2024) employs proximal gradient terms to estimate perturbation and suggests that applying perturbation only to the final layer is sufficient. FedGAMMA (Dai et al. 2023) enhances FedSAM by integrating variance reduction of Scaffold (Karimireddy et al. 2020). FedSpeed (Sun et al. 2023b) optimizes the FL process using the Alternating Direction Method of Multipliers (ADMM). Additionally, FedSMOO (Sun et al. 2023a) noticed that local perturbation guides client models toward their respective local flat minima, leading to model inconsistency, therefore, FedSMOO employs two ADMM methods to correct both local updates and perturbation. FedLESAM (Fan et al. 2024) uses a global perturbation parallel to the global gradient direction as a local perturbation estimate for the client, thereby allowing local clients to perceive the sharpness of the global model. FedMRUR (An et al. 2024) utilizes the hyperbolic graph fusion technique to mitigate model inconsistency. Moreover, SAM also has been applied in decentralized federated learning (Shi et al. 2023; Li et al. 2023).

However, these methods often overlook the impact of different perturbation amplitude and the selection of parameters to perturb from the perspective of sharpness. Given the significant influence of SAM's perturbation amplitude on convergence speed and accuracy, it is essential to develop a reasonable perturbation strategy in PFL.

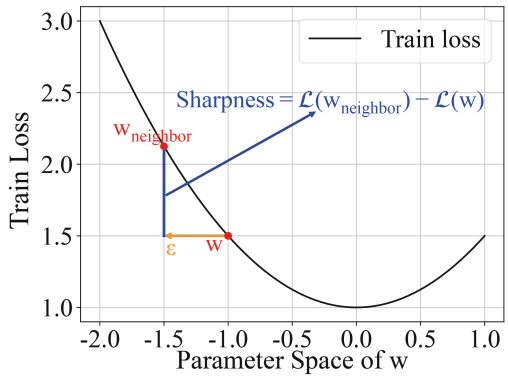


Figure 1: Understanding sharpness intuitively.

3 Preliminaries

In this section, we introduce the problem definition of PFL and then explain the concept of SAM.

3.1 Problem Definition of PFL

Unlike traditional federated learning, for a system with N clients, PFL trains a corresponding personalized local model W_i for each client i ($i \in [1, N]$) on their local data $D_i = \{D_i^{train}, D_i^{test}\}$. We use $\mathcal{L}_i(W_i; D_i)$ to denote the loss function of client i. The PFL goal is formulated as follows:

$$\min_{W_1, W_2, \ldots, W_N} \sum_{i=1}^{N} \mathcal{L}_i(W_i; D_i), \tag{1}$$

Each client minimizes its loss $\mathcal{L}_i(W_i; D_i^{train})$ with local data D_i^{train}. Due to the limited local data of each client, the model is prone to overfitting easily, resulting in poor generalization performance of W_i on D_i^{test}. The primary challenge in PFL is to mitigate the impact of non-IID data and improve the generalization performance of clients on local data during client collaboration.

3.2 Sharpness-Aware Minimization

SAM has shown strong generalization capabilities in centralized learning and is promising to alleviate the non-IID data challenge in PFL. The objective of SAM is to minimize the loss function and smooth the loss landscape by solving the following min-max problem:

$$\min_{W} \max_{\|\boldsymbol{\varepsilon}\|_2 \leq \rho} \mathcal{L}(W + \boldsymbol{\varepsilon}), \tag{2}$$

where W is the model weights, and $\boldsymbol{\varepsilon}$ is the perturbation added when solving the maximization problem, constrained by the perturbation amplitude ρ. The sharpness of $\mathcal{L}(W)$ in the neighborhood of W, i.e., $U(W, \boldsymbol{\varepsilon}) = \{W' \mid |W' - W| < \boldsymbol{\varepsilon}\}$, is defined as follows:

$$\max_{\|\boldsymbol{\varepsilon}\|_2 \leq \rho} \mathcal{L}(W + \boldsymbol{\varepsilon}) - \mathcal{L}(W). \tag{3}$$

Fig. 1 provides an intuitive display of the sharpness of the training loss in the neighborhood $U(W, \boldsymbol{\varepsilon})$. The smaller the sharpness, the flatter the loss landscape. Eq. (2) can be considered as simultaneously minimizing the sharpness and loss in the training space.

$$\min_{W}[\max_{\|\boldsymbol{\varepsilon}\|_2 \leq \rho} \mathcal{L}(W + \boldsymbol{\varepsilon}) - \mathcal{L}(W) + \mathcal{L}(W)]. \tag{4}$$

SAM uses the first-order Taylor expansion and the dual norm to obtain an approximate solution to the perturbation:

$$\boldsymbol{\varepsilon} = \underset{\|\boldsymbol{\varepsilon}\|_2 \leq \rho}{\operatorname{argmax}} \mathcal{L}(W + \boldsymbol{\varepsilon}) \approx \rho \cdot \frac{\nabla_W \mathcal{L}(W)}{\|\nabla_W \mathcal{L}(W)\|_2}. \tag{5}$$

Since $\frac{\nabla_W \mathcal{L}(W)}{\|\nabla_W \mathcal{L}(W)\|_2}$ is the gradient direction at W, the internal maximization problem can be regarded as a one-step gradient ascent at W, with ρ being the ascent step size. Finally, the gradient $\nabla_W \mathcal{L}(W + \boldsymbol{\varepsilon})$ at the perturbed parameter is used to solve the external minimization problem through a one-step gradient descent to actual update parameter W.

4 Flexible Sharpness-Aware Method

In this section, we introduce our FedFSA, which includes the motivation of FedFSA, the perturbation sensitivity metric to estimate layer-wise sharpness, and the application of perturbation sensitivity in PFL. Finally, we provide the FedFSA procedure in detail.

4.1 Motivation

Inspired by DISAM (Zhang et al. 2024) and FedLESAM (Fan et al. 2024), which suggest that larger perturbations may improve generalization but slow convergence, and by SSAM (Mi et al. 2022) and SAMON (Mueller et al. 2024), which indicate that not all parameters need perturbation, we hypothesize that the slow convergence caused by large perturbations may result from applying these perturbations to unnecessary parameters. Based on this, we designed FedFSA, which flexibly selects parameters to apply large perturbations according to their sharpness relative to the global model.

It should be noted that typically, the sharpness calculation requires the Hessian trace of the parameters (Ahn, Jadbabaie, and Sra 2024), which is unacceptable in resource-constrained FL environments. Additionally, sharpness can be computed using the defined formula in Eq. (3), but this approach does not permit layer-wise sharpness calculations. Due to the lack of a suitable method for calculating sharpness in FL, we first design a metric, perturbation sensitivity, to estimate the layer-wise sharpness.

4.2 Perturbation Sensitivity

Given a model W with L layers whose parameter set is expressed as $W = \{w_1, \ldots, w_k, \ldots, w_L\}$. By Eq. (3), we use $\mathcal{L}_{\text{SAM}}(W) = \mathcal{L}(W + \boldsymbol{\varepsilon})$ to simplify the maximum loss in the neighborhood of W. Therefore, the sharpness of W can be simplified to:

$$\mathcal{L}_{\text{SAM}}(W) - \mathcal{L}(W). \tag{6}$$

Inspired by parameter sensitivity (Lee, Ajanthan, and Torr 2018; Molchanov et al. 2019; Zhang et al. 2022; Wu et al. 2023), we associate sharpness with perturbation and propose

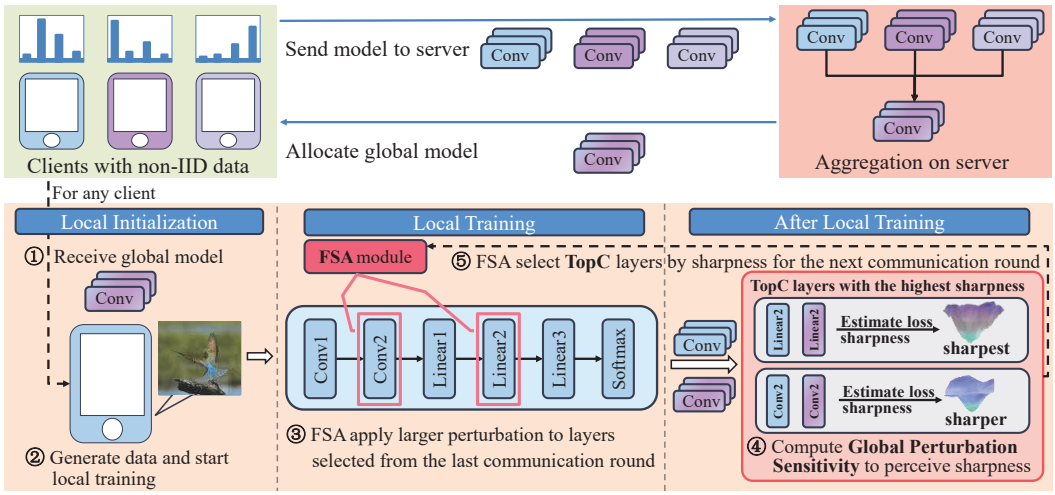


Figure 2: The workflow of FedFSA. (1) Each client receives the global model; (2) Clients begin to train the model using their local data; (3) During local training, FSA applies larger perturbation to layers selected from the last communication round; (4) After training, FSA estimates sharpness by computing perturbation sensitivity using the locally trained model and the initial global model, and then ranks the TopC layers with the highest sharpness; (5) Based on the ranking, FSA selects the layers with the highest sharpness for application in the next communication round.

the concept of perturbation sensitivity as an approximation of sharpness. We define the perturbation sensitivity of the k-th layer as the degree of variation in the model output or loss function after removing the perturbation of the k-th layer parameter w_k. That is, let $\boldsymbol{\varepsilon} = \{\epsilon_1, \ldots, \epsilon_k, \ldots, \epsilon_L\}$ be the perturbation of each layer, the perturbation sensitivity of the k-th layer s_k can be expressed as:

$$s_k = |\mathcal{L}(W + \boldsymbol{\varepsilon}) - \mathcal{L}(w_1 + \epsilon_1, \ldots, w_k, \ldots, w_L + \epsilon_L)|. \tag{7}$$

However, as can be seen in Eq. (7), evaluating the layer-wise perturbation sensitivity requires an additional forward propagation, which seriously increases the computational overhead. To solve this problem, we use the first-order Taylor approximation to substitute the calculation of s_k:

$$\begin{aligned} s_k &= |\mathcal{L}(W + \boldsymbol{\varepsilon}) - \mathcal{L}(w_1 + \epsilon_1, \ldots, w_k, \ldots, w_L + \epsilon_L)| \\ &= |\nabla_{w_k} \mathcal{L}(W + \boldsymbol{\varepsilon}) \cdot \epsilon_k + R_1(W + \boldsymbol{\varepsilon})| \\ &\approx |\nabla_{w_k} \mathcal{L}(W + \boldsymbol{\varepsilon}) \cdot \epsilon_k|. \end{aligned} \tag{8}$$

The gradients of parameters can be obtained by one back propagation, which reduces the computation overhead of s_k.

4.3 Global Perturbation Sensitivity

In FL scenarios, clients typically update the model locally for several epochs. To reduce computation cost, a straightforward way to apply Eq.(8) to FL is to compute the perturbation sensitivity only in the final local epoch. However, the fluctuation of one iteration ignores the information of sharpness accumulated throughout the entire local training process, which may lead to suboptimal results. To overcome this limitation, we introduce the global perturbation sensitivity metric, which estimates the sharpness of the clients relative to the global model by measuring the variation in loss before and after local training, thereby perceiving sharpness in a wide range of neighborhoods.

Therefore, we denote the loss of the global model at the local initialization phase as $\mathcal{L}(W_g)$, and the loss at the end of the local training phase of client i as $\mathcal{L}_i(W_i)$. The variation in parameters $W_g - W_i$ during training can be considered as a large perturbation $\boldsymbol{\varepsilon}_i$, added to the local model to reasonably explore the neighborhood, i.e., $W_g = W_i + \boldsymbol{\varepsilon}_i$ and $\mathcal{L}_{\text{SAM}}(W_i) = \mathcal{L}_i(W_g)$. Consequently, the global perturbation sensitivity of clients is defined as follows:

$$\begin{aligned} s_i &= |\mathcal{L}_{\text{SAM}}(W_i) - \mathcal{L}_i(W_i)| \\ &= |\mathcal{L}_i(W_i + \boldsymbol{\varepsilon}_i) - \mathcal{L}_i(W_i)|. \end{aligned} \tag{9}$$

Similarly to Eq. (8), the global perturbation sensitivity of the k-th layer of the client i is:

$$s_{i,k} \approx |\nabla_{w_{i,k}} \mathcal{L}_i(W_i + \boldsymbol{\varepsilon}_i) \cdot \epsilon_{i,k}|. \tag{10}$$

Specifically, in Eq. (10), we replace $\nabla_{w_{i,k}} \mathcal{L}_i(W_i + \boldsymbol{\varepsilon}_i)$ with the variation of parameter $\Delta w_{i,k} = w_{i,k}^{t,E} - w_{i,k}^{t,0}$. Here, $w_{i,k}^{t,e}$ represents the k-th layer parameter in communication round $t \in [1, T]$ for client model W_i after the e-th local iteration, $e \in [0, E]$. The actual value of the perturbation $\epsilon_{i,k} = w_{i,k}^{t,0} - w_{i,k}^{t,E}$. Therefore, $s_{i,k}^t$ can be ultimately computed by Eq. (11):

$$\begin{aligned} s_{i,k}^t &\approx |\Delta w_{i,k} \cdot \epsilon_{i,k}| \\ &= |(w_{i,k}^{t,E} - w_{i,k}^{t,0}) \cdot \epsilon_{i,k}| \\ &= (w_{i,k}^{t,E} - w_{i,k}^{t,0})^2. \end{aligned} \tag{11}$$

Since the perturbation sensitivity only needs to be calculated once during the entire local training process, and $w_{i,k}^{t,E} - w_{i,k}^{t,0}$ can be directly used for the subsequent momentum calculations, the computation overhead of FedFSA and MoFedSAM is essentially the same.

Algorithm 1: Main Process of FedFSA

Input:communication round T, local iterations E, perturbation amplitude ρ_{larger} and ρ_{default}, FSA select $Top[C]$ layers, global and local learning rate η_g, η_l, momentum coefficient α, the number of local updates K, momentum for clients Δ.

Output: Personalized model W_i^T for each client

for $t = 1, 2, \ldots, T$ **do**
  server select active clients set S^t at round t
  for client $i \in S^t$ parallel **do**
    client i receive server W_g^t and set $W_i^{t,0} = W_g^t$
    for $e = 1, 2, \ldots, E$ **do**
      if $w_{i,k}^{t,e}$ is selected, i.e., $k \in Top_i[C]$ **then**
        apply larger perturbation $\epsilon_{i,k}^{t,e} = \rho_{\text{larger}} \frac{\nabla \mathcal{L}_i(w_{i,k}^{t,e})}{\|\nabla \mathcal{L}_i(W_i^{t,e})\|}$
      else
        apply default perturbation $\epsilon_{i,\not k}^{t,e} = \rho_{\text{default}} \frac{\nabla \mathcal{L}_i(w_{i,\not k}^{t,e})}{\|\nabla \mathcal{L}_i(W_i^{t,e})\|}$
      end if
      $v_i^{t,e+1} = \alpha \nabla \mathcal{L}_i(W_i^{t,e} + \boldsymbol{\varepsilon}_i^{t,e}) + (1-\alpha)\Delta^t$
      $W_i^{t,e+1} = W_i^{t,e} - \eta_l v_i^{t,e+1}$
    end for
    Parameter variation after local update $\Delta_i^t = W_i^{t,E} - W_i^{t,0}$
    compute perturbation sensitivity $s_i^t = (\Delta_i^t)^2$ and select $Top_i[C]$ Layers sorted by s_i^t
  end for
  server aggregate $\Delta_g^{t+1} = \frac{1}{|S^t|} \sum_{i \in S^t} \Delta_i^t$
  Update global parameter $W_g^{t+1} = W_g^t + \eta_g \Delta_g^{t+1}$
  momentum for clients $\Delta^{t+1} = \frac{1}{\eta_l K} \Delta_g^{t+1}$
end for
return $[W_1^T, W_2^T, ..., W_N^T]$

4.4 Flexible Sharpness-Aware Procedure

During the gradient ascent step in SAM on the client-side local learning process, FedFSA applies larger perturbation to critical layers that can represent the entire model. To identify critical layers, We use the global perturbation sensitivity metric. The TopC most sensitive layers are selected to employ larger perturbation in the next communication round. By customizing the SAM optimizer for each client, FedFSA mitigates the impact of non-IID data and enhances the generalization performance of local clients. The details of the FedFSA process are described in Algorithm 1.

Initialization In each communication round $t \in [1, T]$, client $i \in [1, N]$ trains E iterations minimizing its local loss function (e.g., Cross-Entropy loss for image classification tasks) to update the personalized model $W_i^t = \{w_{i,1}^t, \ldots, w_{i,k}^t, \ldots, w_{i,L}^t\}$. The $Top_i[C]$ list of each client i is initialized to empty, and C is a hyperparameter of FedFSA to control the number of critical layers to enlarge perturbation. The larger and default perturbation amplitude ρ_{larger} and ρ_{default}. The learning rates η_g, η_l for global and local model updates. The momentum coefficient α and the number of local updates K.

Training In the t-th communication round, we randomly select an activated client set S^t according to the client participation rate and send the current global model W^t to all activated clients. Lines.6-Lines.10, client i applies a larger perturbation to the TopC critical layers selected in the previous round. Lines.11-Lines.12 Client i uses the perturbed parameters to calculate the actual gradient and combines it with the global gradient for momentum update. After the local training is completed, Lines.15 client i evaluates the TopC critical layers in preparation for the next training. Lines.17-Lines.19, the server aggregates the parameter variation Δ_i^t of all participating clients, updates the global model, and calculates the momentum Δ^{t+1} for the client in the next round.

Details When evaluating critical layers, we follow the LLMC (Adilova et al. 2024), considering only the weight parameters of the model's convolutional or fully connected layers, such as conv.weight and fc.weight, and excluding non-weight layers, such as conv.bias, fc.bias, and batch normalization layers (both bn.weight and bn.bias).

5 Experiment

5.1 Experiment Setting

Dataset and partition strategy We evaluate our FedFSA using four public image classification benchmark datasets: Fashion-MNIST (FMNIST) (Xiao, Rasul, and Vollgraf 2017), CIFAR10/100 (Krizhevsky, Hinton et al. 2009), and Tiny-ImageNet (TINY) (Le and Yang 2015). To verify the effectiveness of our method in different non-IID scenarios, we follow (Hsu, Qi, and Brown 2019) to partition the dataset using widely used Dirichlet (Dir) and Pathological (Pat) sampling, see more in Appendix B.1 and B.5. We ensure that each client holds the same amount of data regardless of the partitioning strategy. Specifically, in the Pat partition, for FMNIST and CIFAR10, each client is randomly assigned 5 classes i.e., Pat(s=5), while for CIFAR100 and TINY are Pat(s=15) and Pat(s=50), respectively. In the Dir partition, for FMNIST and CIFAR10, we set Dir(β=0.5), and for the rest, we set Dir(β=0.3). After partitioning, the data is split into 70% training and 30% testing sets.

Model architecture We follow the (Zhang et al. 2023a) and use a 5-layer Convolutional Neural Network (CNN) model similar to LeNet5 (Lecun et al. 1998). For the specific configurations, please refer to Appendix B.2.

Implementation details For FL training-related hyperparameters, we set the local learning rate to 0.1 and the global learning rate to 1.0. The number of local epochs is set to 10, and a batch size of 48. The maximum communication round is set to 250 for FMNIST and 500 for other datasets to ensure full convergence, with a 10% participation ratio of a total of 100 clients. At each communication round, we evaluate the average test accuracy obtained from all clients. All experiments are repeated with three random seeds {23, 100, 200}. We employ SGD as the base optimizer for all baselines including SAM-based approaches. The hyperparame-

Method	FMNIST(%)		CIFAR10(%)		CIFAR100(%)		TINY(%)	
	Pat(5)	Dir(0.5)	Pat(5)	Dir(0.5)	Pat(15)	Dir(0.3)	Pat(50)	Dir(0.3)
FedAvg-FT	92.51±0.21	91.59±0.47	81.79±0.61	81.41±0.77	53.70±0.26	41.09±0.14	23.03±0.46	20.75±0.35
FedCR	93.67±0.11	92.43±0.41	84.00±0.14	83.49±0.32	59.91±0.35	42.74±0.39	23.71±0.78	—
FedALA	92.99±0.22	91.83±0.39	81.62±1.08	81.76±0.53	54.92±0.20	39.75±1.07	—	—
FedSAM	93.01±0.11	91.89±0.39	84.16±0.20	83.84±0.44	59.41±0.39	44.66±0.40	28.98±0.15	25.55±0.53
MoFedSAM	**93.79±0.21**	**92.94±0.27**	88.33±0.14	88.16±0.25	70.33±0.29	51.02±2.21	33.80±0.29	28.02±0.60
FedSMOO	93.51±0.05	92.46±0.19	87.22±0.16	87.03±0.24	66.56±0.68	52.34±1.17	26.29±0.02	21.96±0.04
FedSpeed	93.68±0.15	92.88±0.20	88.24±0.37	87.99±0.25	68.24±0.15	52.61±0.55	29.60±0.28	24.92±0.43
FedFSA	93.78±0.11	92.88±0.35	**88.64±0.09**	**88.37±0.27**	**72.28±0.40**	**60.87±0.61**	**41.39±0.25**	**33.76±0.93**

Table 1: Average test accuracy under Pathological and Dirichlet non-IID settings on FMNIST, CIFAR10, CIFAR100, and TINY. Bold fonts highlight the best accuracy.

ters of baselines are set according to their default configurations, as detailed in Appendix B.4, except for MoFedSAM, where we make specific adjustments to the perturbation amplitude as needed. For FedFSA, we set momentum coefficient α to 0.1, the perturbation amplitude ρ_{default} to 0.05 (0.1 for FMNIST and CIFAR10) and ρ_{larger} to 0.9 (0.2 for FMNIST and CIFAR10). Additionally, we set $\text{Top}C$ to 2.

Baselines The proposed method, FedFSA, is compared first with the classic FedAvg (McMahan et al. 2017) and then with two recent PFL methods representing two types: FedCR (Zhang et al. 2023a), which effectively utilizes shared representations between clients by minimizing the difference in local and global conditional mutual information, representing parameter decoupling; and FedALA (Zhang et al. 2023b), which adaptively aggregates the global model and local model towards the local objective, representing model interpolation. To demonstrate the superiority of reasonably enlarging perturbation in FedFSA, we also compare it with recent federated SAM-based approaches, such as FedSAM (Caldarola, Caputo, and Ciccone 2022), MoFedSAM (Qu et al. 2022), FedSpeed (Sun et al. 2023b), and FedSMOO (Sun et al. 2023a).

5.2 Performance Evaluation.

Due to the local overfitting problem of PFL methods, we take the best accuracy in each experiment. To ensure fairness for all methods, especially for non-PFL methods, we further tune the model classifiers of all clients using SGD in one round after training.

Comparison with baselines As shown in Table 1, FedFSA achieves state-of-the-art performance in most cases, except for FMNIST. Specifically, on the CIFAR100 dataset, FedFSA achieves 60.87% in the Dir(0.3) setups, which is 8.26% higher than the second-highest accuracy. The more complex the dataset, the more pronounced the effectiveness of FedFSA. We attribute this to the fact that the model trained on simpler datasets has a flatter loss landscape and is easier to converge to a global minimum, thus larger perturbation has limited impact on improving generalization performance. In contrast, for more complex datasets, larger perturbation can be effective in escaping local sharp minima. For unknown reasons, FedCR and FedALA failed to converge on TINY, which is marked as — in the table.

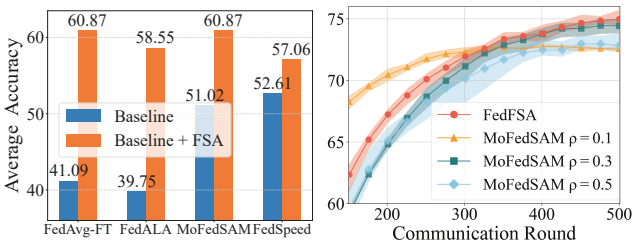
Impact of heterogeneity and scalability To assess the effectiveness of FedFSA under varying degrees of heterogeneity, we adjust the β in Dir(β) on CIFAR100. In PFL settings, a smaller β indicates greater heterogeneity, which typically results in higher test accuracy. In Table 2, As the heterogeneity decreases, the performance of all methods declines, but FedFSA consistently maintains a performance lead. To show the scalability of FedFSA, we also conduct two experiments with 50 and 100 clients in the Dir(0.3) setting on TINY. Given that the total amount of data for TINY remains constant across experiments, increasing the number of clients results in each client receiving fewer data, which exacerbates the scarcity of local data. Therefore, as illustrated in Table 2, when the number of clients increases to 100, the accuracy of all methods decreases to varying extents. Although the accuracy of FedFSA drops by about 3%, it still performs the best.

Method	Heterogenity		Scalability	
	CIFAR100(%)		TINY(%)	
	Dir(0.1)	Dir(1)	50 clients	100 clients
FedAvg-FT	52.92±0.46	34.58±0.50	22.24±0.39	20.75±0.35
FedCR	55.87±0.44	30.31±0.36	28.62±0.39	—
FedALA	52.56±1.32	34.72±0.63	—	—
FedSAM	58.42±1.08	37.40±0.24	26.73±0.12	25.55±0.53
MoFedSAM	66.79±0.75	44.41±0.59	28.56±0.38	28.02±0.60
FedSMOO	65.85±0.71	46.58±0.52	25.87±0.27	21.96±0.04
FedSpeed	65.98±0.91	46.19±0.56	31.60±0.25	24.92±0.43
FedFSA	**67.73±0.69**	**52.49±0.35**	**36.90±0.43**	**33.76±0.93**

Table 2: Average test accuracy at different levels of heterogeneity on CIFAR100 and scalability with different numbers of clients on TINY.

5.3 Applicability Evaluation

Method applicability As the FSA only modifies local optimizer, it can be applied to most existing FL methods. We apply FSA to baselines without modifying other learning processes to evaluate its effectiveness except for FedCR and FedSMOO. This exception might be because FedCR, a representation learning method, adds a representation layer to the model, and applying excessive perturbation to this layer

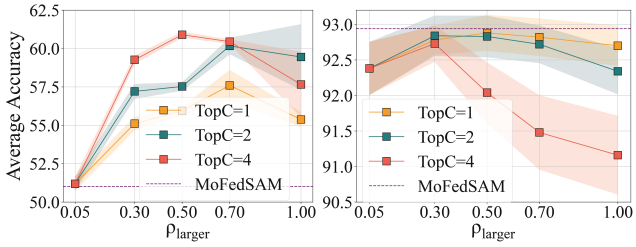


(a) CIFAR100 Dir(0.3) (b) CIFAR100 Dir(0.3)

Figure 3: Average test accuracy to demonstrate the applicability of FSA. (a) shows the applicability of FSA to other FL methods and (b) shows the applicability of FSA to ResNet18.

might interfere with its output. FedSMOO uses perturbation to compute correction, and larger local perturbation might reduce its original effectiveness. We report the accuracy and improvements in Figure 3a on CIFAR100 in the Dir(0.3) setting. Since FedFSA can be seen as FedAvg + FSA + local momentum, we use the same accuracy of 60.87 for FedAvg-FT and MoFedSAM.

Model applicability To verify that our FedFSA still works on a deeper model, we conducted more experiments on ResNet18 (He et al. 2016) with batch normalization. We tune the parameter ρ for MoFedSAM in the range of $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$, and selected the best $\rho = 0.3$ based on the balance between generalization accuracy and convergence speed. Then, we set $\mathrm{TopC} = 5$, $\rho_{\mathrm{default}} = 0.05$, and $\rho_{\mathrm{larger}} = 1$ for FedFSA. As shown in Figure 3b, FedFSA achieves faster convergence and better generalization performance compared to MoFedSAM.

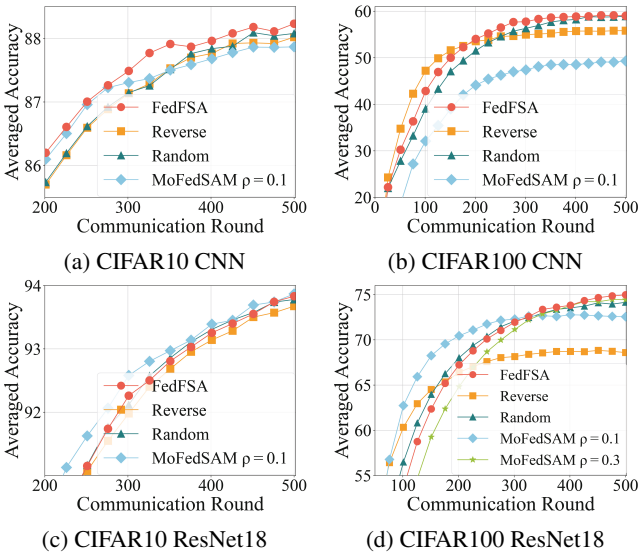


(a) CIFAR100 Dir(0.3) (b) FMNIST Dir(0.5)

Figure 4: The impact of the hyperparameter ρ_{larger} and TopC of FedFSA on different types of datasets, ranging from complex to simple. The perturbation amplitude ρ_{default} for MoFedSAM is set to 0.1, while for FedFSA is 0.05.

5.4 Ablation Evaluation

Effect of hyperparameter ρ_{larger} and TopC Since perturbation amplitude ρ critically influences the convergence and performance of SAM-based FL approaches, here we tune ρ_{larger} and TopC of FedFSA on CIFAR100 and FMNIST. We then compare the average test accuracy with MoFedSAM. As shown in Figure 4, the test accuracy on CIFAR100 initially increases significantly with ρ_{larger}, benefiting from the large perturbation's ability to escape sharp local minima. However, as ρ_{larger} increases, the accuracy declines due to excessive perturbation. FMNIST exhibits the same characteristic; however, no matter how ρ_{larger} and TopC are tuned, it does not achieve the performance of MoFedSAM due to the flatter loss landscape compared to CIFAR100.



(a) CIFAR10 CNN (b) CIFAR100 CNN

(c) CIFAR10 ResNet18 (d) CIFAR100 ResNet18

Figure 5: The effect of different critical parameter selection schemes on CIFAR10 and CIFAR100.

Effect of FSA module We compare three critical parameter selection methods: 'FedFSA' and 'Reverse', which uses FSA to choose perturbation-insensitive layers, and 'Random', which randomly selects TopC layers as critical layers. We conduct experiments on CIFAR10 and CIFAR100 using CNN and ResNet18 models under Dir(0.3) settings. The hyperparameters are consistent with Table 1 and Figure 3b, respectively.

The results are shown in Figure 5. For clarity, error bars are omitted. On the simpler CIFAR10 task, 'FedFSA' with CNN achieves the best accuracy throughout training, while with ResNet18, its results are similar to 'Random' due to the flatter loss landscape. Regardless of the model, 'Reverse' performs the worst. On CIFAR100, 'Reverse' shows faster convergence in the early stages but converges to a poor result later, with this drawback being more pronounced on the deeper ResNet18. In contrast, both 'FedFSA' and 'Random' converge to better results, with 'FedFSA' achieving a higher peak accuracy on ResNet18, indicating the selection scheme of FSA can better mitigate non-IID data issues.

6 Conclusion

In this study, we propose a novel PFL method, FedFSA, to improve the local optimization process by flexibly perceiving sharpness. FedFSA achieves or surpasses the performance of state-of-the-art methods on complex datasets such

as CIFAR100 and under different levels of data heterogeneity, while not introducing much computational overhead. Additionally, we demonstrate the scalability of FedFSA with varying numbers of clients and its applicability to other FL methods and to models of different structures.

Limitations & Broader Impacts We were unable to theoretically analyze the specific impact of FedFSA on improving generalization. Additionally, the perturbation amplitude ρ_{larger} is fixed, and there may be more effective adaptive methods to adjust ρ_{larger}, which is a direction worth exploring in the future.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62376228, the Sichuan Central-Guided Local Science and Technology Development under Grant 2023ZYD0165, and the Chengdu Science and Technology Program under Grant 2023-JB00-00016-GX.

References

Adilova, L.; Andriushchenko, M.; Kamp, M.; Fischer, A.; and Jaggi, M. 2024. Layer-wise linear mode connectivity. In *The Twelfth International Conference on Learning Representations*.

Ahn, K.; Jadbabaie, A.; and Sra, S. 2024. How to Escape Sharp Minima with Random Perturbations. In *Forty-first International Conference on Machine Learning*.

An, X.; Shen, L.; Hu, H.; and Luo, Y. 2024. Federated Learning with Manifold Regularization and Normalized Update Reaggregation. *Advances in Neural Information Processing Systems*, 36.

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

Caldarola, D.; Caputo, B.; and Ciccone, M. 2022. Improving generalization in federated learning by seeking flat minima. In *European Conference on Computer Vision*, 654–672. Springer.

Chen, J.; Li, H.; and Chen, C. P. 2024. Boosting sharpness-aware training with dynamic neighborhood. *Pattern Recognition*, 153: 110496.

Collins, L.; Hassani, H.; Mokhtari, A.; and Shakkottai, S. 2021. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, 2089–2099. PMLR.

Dai, R.; Yang, X.; Sun, Y.; Shen, L.; Tian, X.; Wang, M.; and Zhang, Y. 2023. Fedgamma: Federated learning with global sharpness-aware minimization. *IEEE Transactions on Neural Networks and Learning Systems*.

Fan, Z.; Hu, S.; Yao, J.; Niu, G.; Zhang, Y.; Sugiyama, M.; and Wang, Y. 2024. Locally Estimated Global Perturbations are Better than Local Perturbations for Federated Sharpness-aware Minimization. In *International Conference on Machine Learning*.

Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2021. Sharpness-aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations*.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

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

Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; and Wilson, A. G. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*.

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

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

Keskar, N. S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; and Tang, P. T. P. 2016. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

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

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lee, G.; Jeong, M.; Kim, S.; Oh, J.; and Yun, S.-Y. 2024. FedSOL: Stabilized Orthogonal Learning with Proximal Restrictions in Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12512–12522.

Lee, N.; Ajanthan, T.; and Torr, P. H. 2018. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*.

Li, H.; Xu, Z.; Taylor, G.; Studer, C.; and Goldstein, T. 2018. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31.

Li, Q.; Shen, L.; Li, G.; Yin, Q.; and Tao, D. 2023. Dfedadmm: Dual constraints controlled model inconsistency for decentralized federated learning. *arXiv preprint arXiv:2308.08290*.

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

Liu, Y.; Mai, S.; Cheng, M.; Chen, X.; Hsieh, C.-J.; and You, Y. 2022. Random sharpness-aware minimization. *Advances in Neural Information Processing Systems*, 35: 24543–24556.

Lyu, K.; Li, Z.; and Arora, S. 2022. Understanding the generalization benefit of normalization layers: Sharpness reduction. *Advances in Neural Information Processing Systems*, 35: 34689–34708.

Ma, X.; Zhang, J.; Guo, S.; and Xu, W. 2022. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10092–10101.

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

Mi, P.; Shen, L.; Ren, T.; Zhou, Y.; Sun, X.; Ji, R.; and Tao, D. 2022. Make sharpness-aware minimization stronger: A sparsified perturbation approach. *Advances in Neural Information Processing Systems*, 35: 30950–30962.

Molchanov, P.; Mallya, A.; Tyree, S.; Frosio, I.; and Kautz, J. 2019. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11264–11272.

Mueller, M.; Vlaar, T.; Rolnick, D.; and Hein, M. 2024. Normalization layers are all that sharpness-aware minimization needs. *Advances in Neural Information Processing Systems*, 36.

Neyshabur, B.; Bhojanapalli, S.; McAllester, D.; and Srebro, N. 2017. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30.

Qu, Z.; Li, X.; Duan, R.; Liu, Y.; Tang, B.; and Lu, Z. 2022. Generalized federated learning via sharpness aware minimization. In *International conference on machine learning*, 18250–18280. PMLR.

Qu, Z.; Li, X.; Han, X.; Duan, R.; Shen, C.; and Chen, L. 2023. How to Prevent the Poor Performance Clients for Personalized Federated Learning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12167–12176.

Reddi, S.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečnỳ, J.; Kumar, S.; and McMahan, H. B. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

Shi, Y.; Shen, L.; Wei, K.; Sun, Y.; Yuan, B.; Wang, X.; and Tao, D. 2023. Improving the model consistency of decentralized federated learning. In *International Conference on Machine Learning*, 31269–31291. PMLR.

Si, D.; and Yun, C. 2024. Practical sharpness-aware minimization cannot converge all the way to optima. *Advances in Neural Information Processing Systems*, 36.

Sun, Y.; Shen, L.; Chen, S.; Ding, L.; and Tao, D. 2023a. Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape. In *International Conference on Machine Learning*, 32991–33013. PMLR.

Sun, Y.; Shen, L.; Huang, T.; Ding, L.; and Tao, D. 2023b. Fedspeed: Larger local interval, less communication round, and higher generalization accuracy. *arXiv preprint arXiv:2302.10429*.

Tan, A. Z.; Yu, H.; Cui, L.; and Yang, Q. 2022. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Wu, X.; Liu, X.; Niu, J.; Zhu, G.; and Tang, S. 2023. Bold but cautious: Unlocking the potential of personalized federated learning through cautiously aggressive collaboration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19375–19384.

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

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

Ye, M.; Fang, X.; Du, B.; Yuen, P. C.; and Tao, D. 2023. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3): 1–44.

Zhang, C.; Bengio, S.; and Singer, Y. 2022. Are all layers created equal? *Journal of Machine Learning Research*, 23(67): 1–28.

Zhang, H.; Li, C.; Dai, W.; Zou, J.; and Xiong, H. 2023a. Fedcr: Personalized federated learning based on across-client common representation with conditional mutual information regularization. In *International Conference on Machine Learning*, 41314–41330. PMLR.

Zhang, J.; Hua, Y.; Wang, H.; Song, T.; Xue, Z.; Ma, R.; and Guan, H. 2023b. FedALA: Adaptive Local Aggregation for Personalized Federated Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9): 11237–11244.

Zhang, R.; Fan, Z.; Yao, J.; Zhang, Y.; and Wang, Y. 2024. Domain-Inspired Sharpness Aware Minimization Under Domain Shifts. In *The Twelfth International Conference on Learning Representations*.

Zhang, Y.; Zhang, H.; Wang, S.; Wu, W.; and Li, Z. 2022. PATS: Sensitivity-aware noisy learning for pretrained language models. *arXiv preprint arXiv:2210.12403*.

Zhou, S.; and Li, G. Y. 2023. Federated learning via inexact ADMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.