Flexible Sharpness-Aware Personalized Federated Learning

Anonymous submission

Roadmap of Appendix The Appendix is organized as follows. We list the notations table in Section A. The details of the experimental setting are in Section B.

A Table of Notations

Notations	Description
Indices:	
i	Index for clients ($i \in [1, N]$)
g	Index for global model
k	The k-th layer parameter ($k \in [1, L]$)
t	The t-th rounds ($t \in [1, T]$)
e	Index for e-th local iterations ($e \in [1, E]$)
FL environments:	
D_i	Local dataset of the i-th client,
β	Concentration parameter for Dirichlet partition
s	Shards per client for the Pathological partition
FL algorithms:	
η_l	Local learning rate
η_g	Global learning rate
ρ_{default}	Perturbation amplitude for SAM-based methods
ρ_{larger}	Larger perturbation amplitude for FedFSA
TopC	TopC layers to be selected for FedFSA
α	Client-level momentum
λ	The coefficient of prox-term for FedSpeed and FedSMOO
β_{FedCR}	Hyperparameter for FedCR
$s_{\text{ALA}}, p_{\text{ALA}}$	Hyperparameters for FedALA
Weights:	
W_i	Weight of the i-th client model
$w_{i,k}^{t,e}$	The k-th layer weight of the i-th client model in communication round t at the e-th local iteration
W_g	Weight of the global model
Loss Functions:	
$\mathcal{L}_i(W_i)$	Local objective for the i-th client
$\mathcal{L}_{\text{SAM}}(W)$	SAM inner maximum objective

Table 3: Table of Notations throughout the paper.

B Experimental Setups

The code is implemented by PyTorch 2.2.1, Cuda 12.2, and Python 3.8. The overall code structure is based on FedCR (Zhang et al. 2023a) and FedSMOO (Sun et al. 2023a) with some modifications for simplicity. We use one A100-40GB GPU and four 2080ti-11GB GPU cards but without Multi-GPU training. For our model applicability experiment on ResNet18, we recommend using a GPU with at least 16GB of memory.

B.1 Datasets

To assess our FedFSA, we utilize four different datasets, with the numbers in parentheses representing the sample sizes for training and testing. Figure 6 illustrates the specific contents of the data images.

- **Fashion-MNIST** (Xiao, Rasul, and Vollgraf 2017) (60,000 / 10,000): consists of grayscale images of Zalando's clothing items, each 28x28 pixels, categorized into 10 classes. The data is augmented using Random Cropping, Horizontal Flipping, and Normalization.
- **CIFAR10** (Krizhevsky, Hinton et al. 2009) (50,000 / 10,000): contains a labeled subset of 80 Million Tiny Images for 10 different classes, with each color image sized at 32x32 pixels. The data is augmented using Random Cropping, Horizontal Flipping, and Normalization.
- **CIFAR100** (Krizhevsky, Hinton et al. 2009) (50,000 / 10,000): contains a labeled subset of 80 Million Tiny Images for 100 different classes, with each color image sized at 32x32 pixels. The data is augmented using Random Cropping, Horizontal Flipping, and Normalization.
- **Tiny-ImageNet** (Le and Yang 2015) (100,000 / 10,000): is a downsized subset version of ImageNet, where images are scaled down to 64×64 pixels. It consists of 200 classes, each with 500 training images and 50 validation images per class. The data is augmented using Random Cropping, Horizontal Flipping, and Normalization.

Each dataset can be publicly accessible through the following sources:

- Fashion-MNIST, CIFAR10, and CIFAR100: Available through the torchvision library.
- Tiny-ImageNet: `http://cs231n.stanford.edu/tiny-imagenet-200.zip`.

Methods	Dataset	Selected	Searched Candidates
FedAvg	all	None	None
FedCR	all	$\beta_{\text{FedCR}} = 0.001$	$\beta_{\text{FedCR}} = \{0.0001, 0.0005, 0.001\}$
FedALA	all	$s_{\text{ALA}} = 80$, $p_{\text{ALA}} = 2$	None
FedSAM	all	$\rho_{\text{default}} = 0.05$	None
MoFedSAM	all	$\rho_{\text{default}} = 0.1$, $\alpha = 0.1$	$\rho_{\text{default}} \in \{0.05, 0.1, 0.3, 0.5\}$
FedSMOO	all	$\rho_{\text{default}} = 0.1$, $\lambda = 0.1$	$\rho_{\text{default}} \in \{0.05, 0.1\}$
FedSpeed	all	$\rho_{\text{default}} = 0.1$, $\lambda = 0.1$	$\rho_{\text{default}} \in \{0.05, 0.1\}$
FedFSA	FMNIST, CIFAR10	$\rho_{\text{default}} = 0.1$, $\rho_{\text{larger}} = 0.2$, TopC = 2, $\alpha = 0.1$	$\rho_{\text{default}} \in \{0.05, 0.1\}$, TopC $\in \{1, 2, 4\}$
	CIFAR100, TINY	$\rho_{\text{default}} = 0.05$, $\rho_{\text{larger}} = 0.9$, TopC = 2, $\alpha = 0.1$	$\rho_{\text{larger}} \in \{0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$

Table 4: Algorithm-specific hyperparameters.

Figure 6: Examples from four datasets

B.2 Model Architecture

In our primary experiments on four benchmarks, we use a 5-layer CNN similar to LeNet5 (Lecun et al. 1998), and its details are listed in Table 5. We also conducted further experiments on ResNet18 (He et al. 2016) to evaluate the model applicability of our FedFSA.

Layer	FMNIST	CIFAR10/100	TINY
1	Conv(3, 4, 5) ReLU MaxPool(2,2)	Conv(3, 64, 5) ReLU MaxPool(2,2)	Conv(3, 32, 5) ReLU MaxPool(2,2)
2	Conv(4, 12, 5) ReLU MaxPool(2, 2)	Conv(64, 64, 5) ReLU MaxPool(2, 2)	Conv(32, 32, 5) ReLU MaxPool(2,2)
3	FC(192, 1024)	FC(1600, 1024)	FC(5408, 512)
4	FC(1024, 1024)	FC(1024, 1024)	FC(512, 1024)
5	FC(1024, 10)	FC(1024, 10/100)	FC(1024, 200)

Table 5: The model architecture of four benchmark experiments. For the convolutional layer (Conv), we list parameters with a sequence of input and output dimensions, and kernel size. For the max pooling layer (MaxPool), we list kernel and stride. For the fully connected layer (FC), we list the input and output dimensions.

B.3 Learning Setups

We employ SGD as the base optimizer for all baselines including SAM-based approaches with an initial local learning rate of 0.1 and the global learning rate of 1.0. The number of local epochs is set to 10, and a batch size of 48. As we are assuming a synchronized FL scenario, we simulate the parallel distributed learning by sequentially conducting local learning for the sampled clients and then aggregating them into a global model with a 10% participation ratio of a total of 100 clients. The standard deviation is measured over 3 random seeds $\{23, 100, 200\}$. The detailed learning setups for each dataset are provided in Table 6.

Datasets	Comm. Rounds	Dir(β)	Pat(s)
FMNIST	250	0.5	5
CIFAR10	500	0.5	5
CIFAR100	500	0.3	15
TINY	500	0.3	50

Table 6: Learning setups

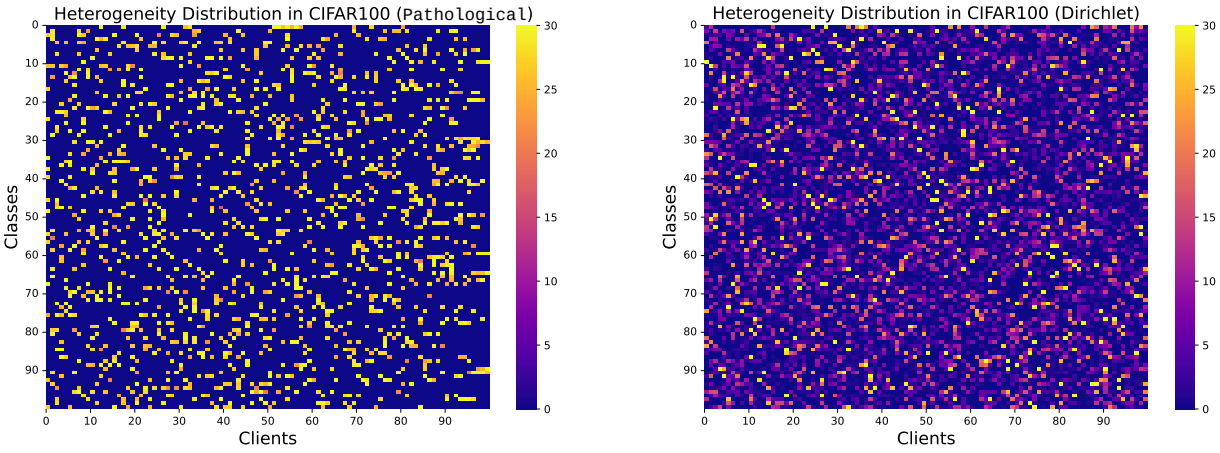
B.4 Algorithm-specific Hyperparameters

We search for hyperparameters and select the best among the candidates. The hyperparameters for each method are provided in Table 4. Please note that for FedSpeed and FedSMOO, the learning rate decays with a factor of 0.998, consistent with the configuration used in their open-source implementations. Additionally, for FedFSA on the TINY dataset, we set the learning rate decay to 0.998 as well. For our model applicability experiment, we set $\rho_{\text{default}} = 0.05$, $\rho_{\text{larger}} = 1$, and TopC = 5.

B.5 Non-IID Partition Strategy

To comprehensively address the issue of data heterogeneity in federated learning, we employ two distinct data partition strategies. Regardless of the partitioning method, we ensure that each client receives an equal amount of data. After partitioning, the data is split into 70% training and 30% testing sets.

- **Pathological Partition**: Only selected categories are sampled with a non-zero probability, resulting in local



(a) Pathological(15) on CIFAR100

(b) Dirichlet(0.3) on CIFAR100

Figure 7: Heat-map of the Pathological and Dirichlet partition.

datasets that follow a uniform distribution of active categories. The data is sorted by label, divided into shards of equal size, and then distributed to clients. The level of heterogeneity increases as the number of shards per client decreases, and vice versa. As illustrated in Figure 7a, categories are mainly represented by either blue or yellow.

- **Dirichlet Partition**: Each category can be sampled with a non-zero probability, and the local datasets follow a Dirichlet distribution. The data samples of class c are allocated to each client i with a probability p_c, where $p_c \approx \text{Dir}(\beta)$. The heterogeneity level increases as the concentration parameter β becomes smaller, and vice versa. As shown in Figure 7b, each category has different colors.

The specific configurations are listed in Table 6.