FedAWA: Adaptive Optimization of Aggregation Weights in Federated Learning Using Client Vectors

Supplementary Material

A. Experiment details

In this section, we provide the details of the experimental setup, environment, datasets, and model architectures used in this paper.

A.1. Client Vector and Local Data.

In Section 4.2, we demonstrated experimentally the relationship between the client vector and local data. In this section, we will provide a more detailed explanation of the experimental setup and offer a further analysis of the results. The Experiments were conducted to verify whether the client vector reflects information about the local data. In the experiment, we set up 12 clients, and the dataset we used was CIFAR-10. To observe the data differences more intuitively, we introduced an extreme scenario of data heterogeneity: The local data of clients 1 to 4 contained only the first 5 classes of the CIFAR-10, clients 5 to 8 had only the left 5 classes, and clients 9 to 12 had local datasets that included all classes. The size of the local dataset for each client was the same. We calculated the distances between the client local datasets via optimal transport [3], and the results are displayed in Figure 2a.

Then, we conducted federated learning training, during which each client obtained its own client vector τ_k after local training. We then compared the distance between the client vectors, with the results displayed in Figure 2b. The relationships between the client vectors closely resemble those of the local data distributions. For example, client 1's client vector exhibits minimal differences with clients 2-4 due to their similar local data distributions. However, the differences between client 1 and clients 5-8 are much larger because of their highly divergent data distributions: client 1's local data contains only the first 5 classes, while clients 5-8 have only the last 5 classes. The differences between client 1 and clients 9-12 are smaller than those with clients 5-8, as clients 9-12 include data from all classes, making their distribution relatively closer to client 1. This demonstrates that the client vector can effectively capture relevant information about the local data. Hence, we explored the possibility of leveraging this phenomenon to enhance the model aggregation process in federated learning.

If the distance is computed directly using the overall model parameters, the results, as shown in Figure 2c, indicate that the distances between models are relatively similar. This is because each client model is optimized from the same global model, and the parameter variations are small relative to the overall model parameters. As a result, the local models do not exhibit significant differences after training, making it difficult to effectively capture the relationships among the local datasets. It is important to note that for both Figure 2b and Figure 2c, we compute the distance between clients vectors and model parameters using 1 - cosine similarity. This method normalizes the values to a consistent scale (ranging from 0 to 2), allowing for a more intuitive comparison of the differences between the models.

A.2. Aggregation Weights.

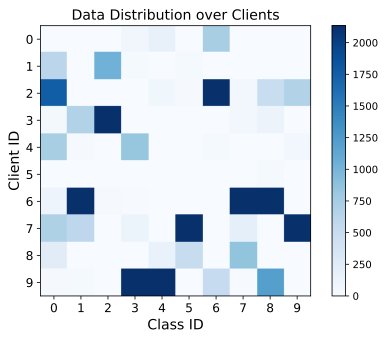
In this section, we provide more details regarding the experiment in Figure 7. We first calculate the similarity between the local dataset and the global dataset, where the partitioning of the local dataset is the same as in Appendix A.1, and the global dataset is the union of all local datasets. We use a pre-trained ResNet20 to extract features from the datasets and compute the distance between the two datasets using Optimal Transport [3]. Since our goal is to measure the similarity between datasets, we convert the OT distance into a similarity score as:

$$\text{Similarity}(P, Q) = \frac{1}{1 + d_{OT}(P, Q)}, \tag{5}$$

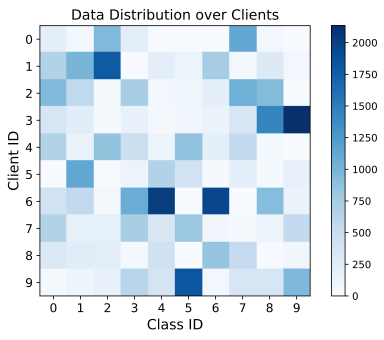
where P and Q represent the distributions of local data and global data, respectively, while $d_{OT}(\cdot, \cdot)$ denotes the optimal transport distance. This results in a k-dimensional dataset vector representing the similarity between each local dataset k and the global dataset. This vector depends solely on the datasets and remains fixed throughout training. The k-th element in the vector indicates the similarity between the local dataset k and the global dataset. We use this as the ideal aggregation weights, assigning higher aggregation weights to datasets more similar to the global dataset, and vice versa [43]. We then evaluate the aggregation weights of different methods by calculating the cosine similarity between the aggregation weights and the data vector. As shown in Figure 7, the results demonstrate the effectiveness of our method.

A.3. Datasets

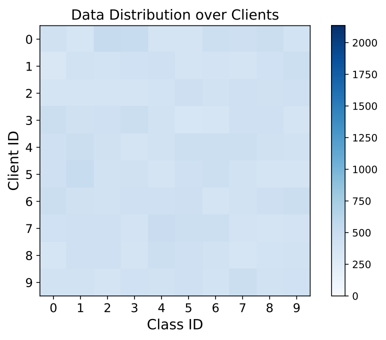
In the experiment, we utilized four image classification datasets: CIFAR-10 [16], CIFAR-100 [16], and Tiny-ImageNet [6], which have been widely employed in prior Federated Learning methods [24, 27, 43]. All these datasets are readily available for download online. To generate a non-IID data partition among clients, we employed Dirichlet distribution sampling Dir_α in the training set of each dataset, the smaller the value of α, the greater the non-IID. In our



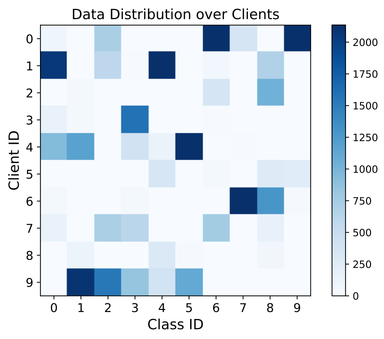
(a) FashionMNIST, $\alpha = 0.1$.



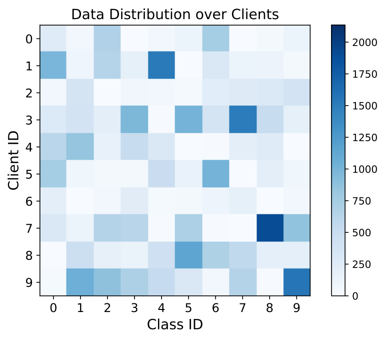
(b) FashionMNIST, $\alpha = 0.5$.



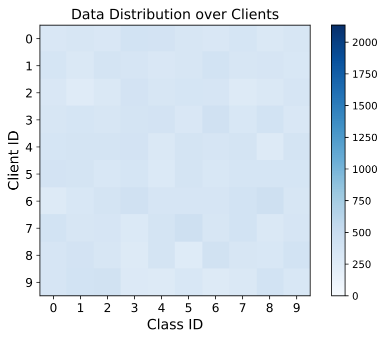
(c) FashionMNIST, $\alpha = 100$.



(d) CIFAR-10, $\alpha = 0.1$.



(e) CIFAR-10, $\alpha = 0.5$.



(f) CIFAR-10, $\alpha = 100$.

Figure 8. Data distribution over categories and clients.

Table 6. Long-tail Scenarios.

Method	**CIFAR10-LT**	**CIFAR100-LT**
FedAvg	77.45	45.87
CReFF	80.71	47.08
CLIP2FL	81.18	48.20
CReFF+AWA(Ours)	**83.11**	**49.63**

Table 7. Multi-domain Scenarios.

Methods	**SVHN**	**USPS**	**MNIST**	**SYN**	**AVG**
FedAvg	76.56	90.85	98.14	55.01	80.14
FedProx	77.01	90.24	98.11	56.66	80.50
FedProto	80.35	92.44	98.30	53.58	81.16
FPL	80.27	92.71	98.31	61.20	83.12
FPL+AWA(Ours)	80.63	91.58	97.76	70.40	**85.09**

Table 8. Text Classification.

Method	**AG News**		**Sogou News**	
	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.5$
FedAvg	73.43	70.37	87.68	91.53
FedProx	65.07	74.56	88.60	92.28
FedAWA	**77.25**	**80.23**	**90.85**	**94.09**

implementation, apart from clients having different class distributions, clients also have different dataset sizes, which we believe reflects a more realistic partition in practical scenarios. We set α =0.1, 0.5, and 100, respectively. When α is set to 100, we consider the data to be distributed in an IID manner. The data distribution across categories and clients is illustrated in Figure 8. Due to the large number of categories, we did not display the data distribution of CIFAR-100 and Tiny-ImageNet. Their distributions are similar to the other two datasets.

A.4. Experiment

In this section, we conducted additional experiments across long-tail, multi-domain, and text classification scenarios to further evaluate the performance of the proposed model under more scenarios.

To further evaluate the performance of the proposed model under more complex data heterogeneity scenarios, we performed comparisons and integrations with algorithms specifically designed for these challenging scenarios. These experiments focused on two primary settings: global long-tail data distribution and multi-domain data distribution.

For the long-tail data distribution scenario, experimental results are presented in Table 6. In these experiments, the proposed algorithm FedAWA was combined with the CReFF [35] algorithm and compared with federated learning methods designed to address long-tail data distributions, such as CReFF [35] and CLIP2FL [36].

Similarly, for the multi-domain data distribution scenario, the results are shown in Table 7. Here, FedAWA was integrated with the FPL [13] algorithm and compared with federated learning methods specifically tailored for multi-domain distributions, namely FPL [13] and FedProto [37].

The experimental results demonstrate that FedAWA consistently improves model performance even in more complex heterogeneous data environments, thereby confirming the stability and robustness of the proposed method.

To further demonstrate the applicability of our method to textual modalities, we conducted additional experiments on NLP datasets AG News [47] and Sogou News [47] under various data heterogeneity settings. As shown in Table 8, FedAWA consistently outperforms baseline methods in text classification tasks.

A.5. Hyperparameters

If not mentioned otherwise, The number of clients, participation ratio, and local epoch are set to 20, 1, and 1, respectively. We set the initial learning rates as 0.08 and set a decaying LR scheduler in all experiments; that is, in each round, the local learning rate is 0.99*(the learning rate of the last round). We adopt local weight decay in all experiments. We set the weight decay factor as 5e-4. We use SGD optimizer as the clients' local optimizer and set momentum as 0.9.

A.6. Models

For each dataset, all methods are evaluated with the same model architectures for a fair comparison. In Table 1, We use ResNet20 [10] for CIFAR-10 and CIFAR-100, ResNet18 for Tiny-ImageNet. In Table 3, we compare the experimental results of different model architectures. The specific model architectures are as follows:

CNN. The CNN is a convolution neural network model with ReLU activations. In this paper CNN consists of 3 convolutional layers followed by 2 fully connected layers. The first convolutional layer is of size (3, 32, 3) followed by a max pooling layer of size (2, 2). The second and third convolutional layers are of sizes (32, 64, 3) and (64, 64, 3), respectively. The last two connected layers are of sizes (64*4*4, 64) and (64, num_classes), respectively.

ResNet, WRN, DenseNet and ViT. We followed the model architectures used in [7, 18, 24]. The numbers of the model names mean the number of layers of the models. Naturally, the larger number indicates a deeper network. For the Wide-ResNet56-4 (WRN56_4) in Table 3, "4" refers to four times as many filters per layer.