

Modeling Global Distribution for Federated Learning with Label Distribution Skew

Tao Sheng^a, Chengchao Shen^{a,*}, Yuan Liu^a, Yeyu Ou^a, Zhe Qu^b and Jianxin Wang^a

^a*School of Computer Science and Engineering, Central South University, Changsha, China*

^b*Department of Electrical Engineering, University of South Florida, Tampa, USA*

ARTICLE INFO

Keywords:

Federated Learning
Label Distribution Skew
Generative Adversarial Network
Non-Independent and Identically Distributed

ABSTRACT

Federated learning achieves joint training of deep models by connecting decentralized data sources, which can significantly mitigate the risk of privacy leakage. However, in a more general case, the distributions of labels among clients are different, called “label distribution skew”. Directly applying conventional federated learning without consideration of label distribution skew issue significantly hurts the performance of the global model. To this end, we propose a novel federated learning method, named FedMGD, to alleviate the performance degradation caused by the label distribution skew issue. It introduces a global Generative Adversarial Network to model the global data distribution without access to local datasets, so the global model can be trained using the global information of data distribution without privacy leakage. The experimental results demonstrate that our proposed method significantly outperforms the state-of-the-art on several public benchmarks. Code is available at <https://github.com/Sheng-T/FedMGD>.

1. Introduction

With the process of deep learning technology, massive training data play a vital role. However, due to some security or privacy issues in some scenarios, access to data is under strict restriction. The data cannot flow out of the corresponding institution, and the lack of data results in the ineffective training of machine learning models. The above data dilemma significantly limits the performance of the trained model and even leads to serious consequences. For example, Watson, a famous artificial intelligence in the medical field, may cause death once it mistakenly gives a drug. To alleviate the above dilemma, a novel framework for machine learning, federated learning [1] is proposed, which provides more data sources for model training through multi-party collaboration without sharing local private data. However, there is generally a large difference in the distribution among clients, which leads to performance degradation of conventional federated learning methods, named Non-Independent and Identically Distributed (Non-IID). Non-IID is one of the important causes of training bias in federated learning, and it has different expressions in different scenarios. Non-IID scenarios in federated learning are carefully delineated in [2], where data heterogeneity caused by differences in the distribution of labels is called label distribution skew. This is the most common scenario in distributed environments that manifests itself in a different distribution of labels among clients, for example, when the client’s data is limited by factors such as geography, the distribution of labels appears significantly different — pandas are only found in China or zoos.

To this end, several methods [3, 4, 5, 6] are proposed to restrict the differences between local and server models in the parameter space, which have shown success in some applications. However, when the data or labels among clients are highly heterogeneous, the above approaches can not take full use of the information in dispersed data, and degrade the performance of the aggregated global model. Therefore, a potential way to address the label distribution skew issue in federated learning is to model the global data distribution among clients. Most existing global data distribution modeling federated learning methods require some auxiliary information such as proxy dataset [7, 8] or pre-known knowledge [9, 10, 11]. Unfortunately, the requirement of these methods is to directly collect private data or data information from clients, which violates the privacy preserving policy and is not applicable to federated learning.

In this paper, we propose a novel federated learning method, named FedMGD, which allows the global model to obtain global information about data distribution across clients without incurring the privacy leakage issue from clients and thus improves its performance. Specifically, we use the global distribution information obtained from modeling

* Corresponding author.

✉ scc.cs@csu.edu.cn (C. Shen)

ORCID(s): 0000-0003-0249-764X (C. Shen)

to refine the aggregation model, by which we can reduce the differences in distribution among clients and thereby mitigate the hazards caused by label distribution skew. In addition, to ensure that the learned global information generated is more similar to the real data, we constrain by both image authenticity and semantic authenticity, which makes FedMGD still maintain high performance in some noisy datasets. To achieve our design goal, our federated learning scheme follows a two-stage procedure. The advantage of this design is to avoid the inability of the model to accurately describe the local label distribution information of the client after federated learning aggregation. In the first stage, we adopt a federated Generative Adversarial Network (GAN) framework, where the generator in the server generates samples as the ones from clients and the discriminator in the corresponding client distinguishes generated samples from the local ones. After the training of federated GAN accomplishes, the generator can effectively capture global information about data distribution across clients. However, directly aggregating the generator parameters used by the federated GAN is also affected by the label distribution skew, which is not conducive to modeling the global distribution. Therefore, we adapted the architecture of the federated GAN and introduced a Realistic Score to more accurately describe the distribution information of the local data. In the second stage, we first aggregate the local trained models from clients and then refine the aggregated model using the samples synthesized by the above generator. In this way, the aggregated global model can significantly benefit from the global information captured by the generator.

In summary, our main contributions are listed as follows:

- We propose a novel federated learning method, FedMGD, which achieves the global modeling of decentralized data distribution by Realistic Score, and effectively eliminates the performance degradation caused by label distribution skew without privacy leakage.
- The experimental results demonstrate that the proposed FedMGD significantly outperforms the state-of-the-art on several public benchmarks.
- A series of ablation experiments demonstrate that FedMGD can correctly model the global data distribution in the scenario of label distribution skew, and provide an additional high-quality data source for downstream tasks.

2. Related Work

2.1. Federated Learning

Federated Learning (FL) is a distributed machine learning scheme that aims to address data collaboration and protect privacy preserving. It allows multiple participants to train machine learning models collaboratively without exposing local data. Federated learning was first proposed by Google [1] to address the problem of updating models locally on Android phones for users. Since federated learning provides an effective solution to the current "Isolated Data Island" problem [12], it has been gradually used in finance [13], security [14], healthcare [15, 16, 17, 18], recommendation systems [19, 20, 21], and other fields.

However, the collaborative multi-party approach in federated learning introduces new issues, mainly in terms of data privacy security and Non-IID datasets. For data privacy security, it has been shown in [22] that federated learning does not fully guarantee data security, and even only gradient exchange among the participants may cause data privacy leakage. The current popular solutions mainly combine Secure Multi-Party Computation (MPC) and Differential Privacy (DP) to ensure privacy security issues on the federated learning process and results [23, 24, 25]. On the other hand, since most of the real world data are distributed in a Non-IID way, this distribution makes the training of models in federated learning very difficult. [2] provides a detailed division of Non-IID scenarios in federated learning, in which the Non-IID caused by the behaviors and habits of different participants is called Feature distribution skew [26, 27]; and the difference of label distribution among participants due to the limitation of geographical location and other factors, this Non-IID is called Label distribution skew, which is also the problem focused on in this paper.

2.2. Label Distribution Skew in Federated Learning

The label distribution skew in federation learning refers to the differences in the distribution of data labels among the clients participating in federated learning. FedAvg [1] is a classical algorithm for solving federated learning problems, which has attracted wide attention because of its simplicity and low communication cost. It learns knowledge from the decentralized data through the transfer and aggregation of model parameters. Unfortunately, in the scenario of label distribution skew, the performance of FedAvg will drop significantly [7].

Recently, many studies have developed different solutions to solve the problem caused by label distribution skew. Some studies restrict the differences between local and server models in the parameter space. For example, Fed-Prox [3] introduced dynamic regularization to ensure the stability of the model in a highly heterogeneous environment by penalizing updates away from the server model. SCAFFOLD [4] corrects deviation in local updates by introducing additional control variables. However, these approaches focus on reducing the differences between the local and global models without taking full advantage of the information contained in the dispersed data.

Other methods are based on knowledge distillation [9, 10, 11]. By using a small portion of the data collected from the clients to guide global model training, the global model can integrate knowledge from the local and mitigate the negative effects of aggregating different data distributions. In addition to the above methods, there are many studies that also strive to reduce the hazards due to label distribution skew by changing the global target or data distribution. Among them, several studies [5, 6] force the data to obey uniform distribution by modeling the target distribution. Other studies [7, 8, 28] suggest that collecting a small portion of data from clients to build a globally shared dataset can reduce the differences in data distribution across clients and improve the performance of the model on label distribution skew data. However, the prerequisites for collecting data from clients may make this approach infeasible in many scenarios with strict privacy requirements. Therefore, we need a method to mitigate the label distribution skew problem in federated learning by obtaining the same data as the client label distribution without violating user privacy.

To solve the privacy issues associated with data collection in federated distillation methods, Zhu et al. [29] proposed FedGen, a data-free knowledge distillation method. It integrates knowledge from the prediction results of the local model by learning a lightweight generator at the server and then distributes the learned knowledge via broadcast and as an inductive bias to regulate the local training. Although FedGen does not collect data from the client directly, the client needs to count and upload the label information of this participating training during each round of communication with the server. This way of obtaining the label distribution also causes the leakage of local distribution information.

2.3. GAN in Federated Learning

In order to better learn global information from the decentralized data distribution, we introduce a distributed GAN to model the global data distribution, thus alleviating the performance degradation caused by label distribution skew in federated learning.

In the traditional GAN [30], the generator and discriminator learn the data distribution through mutual confrontation. Distributed GAN is another application of GAN in a distributed environment. In federated learning, distributed GAN is mainly active in two aspects: one is how to use GAN to attack the client in order to obtain its private data. In the studies of Hitaj et al. [22] and Wang et al. [31], they used GAN to perform recovery attacks on the client's private data from other clients and the server side, respectively. The other focuses on how to generate higher quality data within the constraints of federated learning environments. For example, McMahan et al. [32] propose to train a generator with different privacy protection levels in federated learning, allowing the model to examine data without direct access to real data. FedGAN [33] was proposed to train GAN by parameter exchange in a federated scenario. Hardy et al. [34] proposed a new GAN structure for a distributed environment. On this basis, Chang et al. [35] supplemented the missing modes of medical images through generation.

However, these methods default to data belonging to IID, which is not suitable for data modeling in label offset scenarios. Recently, Yonetani et al. [36] proposed a strategy to solve the problem of label distribution skew in different clients, called Forgiver-First Update (F2U). Although F2U can be used in learning the distribution of some rare classes, the unsupervised approach used ignores the importance of labeling (semantic) information. In the unsupervised label distribution skew scenario, the generated data is likely to be concentrated in only a few classes, which is not conducive to modeling the global data distribution, and thus the use of labels to constrain the type of generated data is necessary. In addition, unlabeled data bring new challenges for the subsequent training of downstream models.

To address the above issues we propose an update of Realistic Score and apply it to FedMGD. Realistic Score adds a semantic truth restriction on the data under different distributions and can better model the data distribution under label distribution skew.

3. Background and Motivation

3.1. Federated Learning Objective

A Federated learning framework typically include multiple participants who collaboratively train a global consensus model by learning the model locally and periodically communicating with a central server. Suppose that K clients

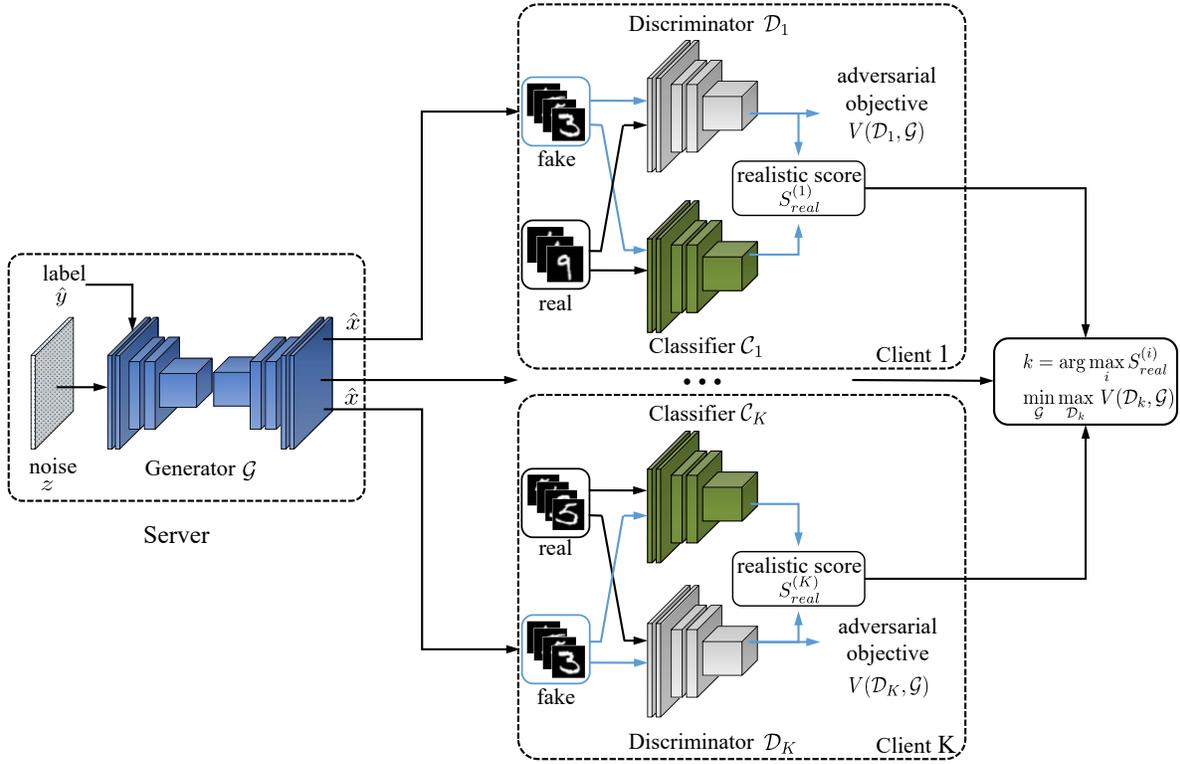


Figure 1: The process of communication between the server and clients in the Generative Adversarial Stage of FedMGD.

are in the federated learning system, and the overall optimization objective is defined as:

$$\min_{\omega} F(\omega) = \sum_{i=1}^K \frac{N_i}{N} F_i(\omega), \quad (1)$$

where ω is the parameter of the global model. For client $i \in [K]$, we define the dataset as $\xi_i = \{(x_m^{(i)}, y_m^{(i)})\}_{m=1}^{N_i}$, where N_i is the number of data samples. As such, the global dataset across clients can be denoted as $\xi = \cup_{i=1}^K \xi_i$, and the total number of participating training data $N = \sum_i N_i$. In each communication round, client i receives model parameters ω from the central server and optimizes the local objective function using local data samples, i.e., $F_i(\omega) = \frac{1}{N_i} \sum_{m=1}^{N_i} \mathcal{L}(\omega; x_m^{(i)}, y_m^{(i)})$, where $\mathcal{L}(\cdot)$ is the local loss function in each client, e.g., cross-entropy. In this paper, we consider that $F_i(\omega^{(t)})$ is non-convex and use a local optimizer such as stochastic gradient descent to process the local training update, which is widely used in existing federated learning works [1, 3]. In the communication round t , client i samples B mini-batches from its dataset ξ_i and performs E local epochs to minimize the objective function $F_i(\omega)$ as:

$$\omega_i^{(t+1)} = \omega_i^{(t)} - \sum_{e=1}^E \eta_i \nabla F_i(\omega_{i,e}^{(t)}; b), \quad (2)$$

where $b \in B$ is one batch in the local epoch e , η_i is the learning rate of local training, and $\nabla F_i(\cdot)$ is the local model gradient.

3.2. Design Motivation

In the traditional distributed IID assumption, the data samples ξ_i are uniformly and randomly distributed among different clients, when $\mathbb{E}_{\xi_i} [F_i(\omega)] = F(\omega)$. However, Non-IID data distribution is a more practical issue caused by different user habits, geographic locations, and other factors, the client optimizes toward different local optimal target F_i . As a result, it leads to a significant degradation or even failure to converge. For two different clients i and j , the

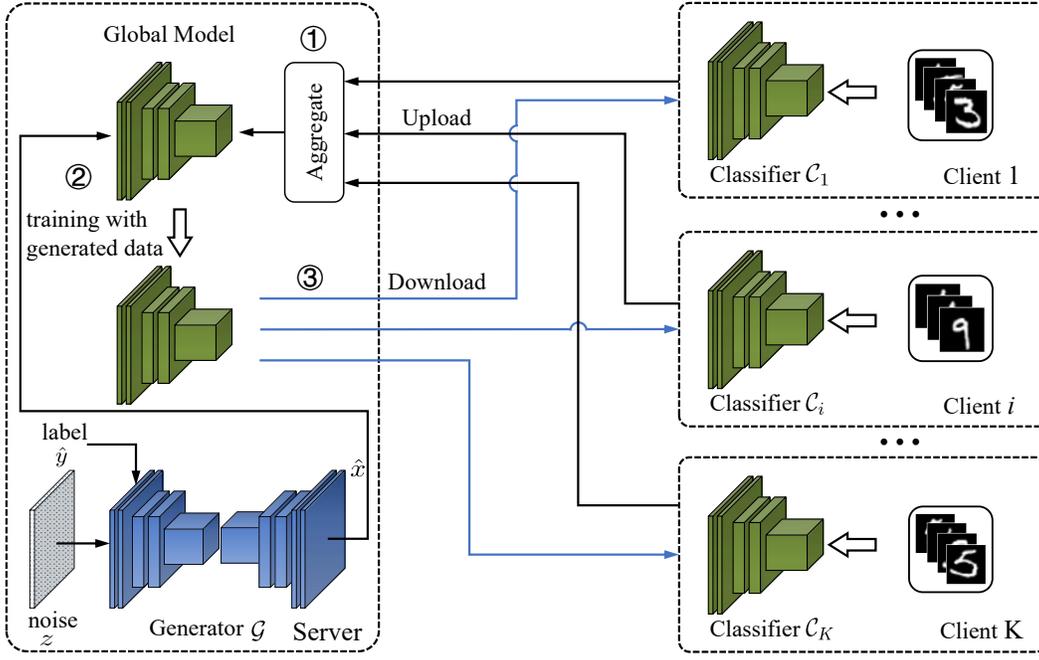


Figure 2: The process of communication between the server and clients in the Federated Enhancement Stage of FedMGD.

input data x and its corresponding label y , $(x, y) \sim p_i(x, y)$ denotes a sample data drawn randomly from the local data distribution in the client i . In formal, for the different two clients i and j , the Non-IID represents $p_i(x, y) \neq p_j(x, y)$. $p(x, y)$ can be written in the form of $p(x|y)p(y)$, and we $p_i(x|y) = p_j(x|y)$ refer to the case of but $p_i(y) \neq p_j(y)$ as label distribution skew.

Some existing works have developed improved algorithms to address the label distribution skew problem, which can be divided into two categories: (1) Designing new aggregation rule [3, 4, 9]. They have been shown the less efficiency and also incur large performance degradation. (2) Modeling the global distribution [7, 8, 28]. They require a globally shared dataset to reduce the client-side label distribution differences. However, this approach relies on the way data is collected from local clients, which makes this approach unusable in scenarios due to the strict privacy policy in federated learning.

According to the above statements, how to model the global distribution without sharing any private data from local clients should be seriously considered. To address this problem, we propose a new federated learning method, named FedMGD, to model the global distribution by introducing a global generative adversarial network to obtain information about the distribution of the global data, which is not required any private data in clients and only manipulate the distribution. We build the generator \mathcal{G} on the server side, which is trained to model the global data distribution across a given client to refine the aggregation model and improve the compatibility of the model between clients.

In particular, in order to avoid the federated learning aggregated model returning wrong local label distribution information and to obtain a well-initialized local model before the federated learning training starts, we divide FedMGD into two stages: the adversarial generation stage and the federated enhancement stage. As shown in Figure 1, in the generative adversarial stage, we adopt the Generative Adversarial Network framework, where generator \mathcal{G} is set on the server side and the discriminators $\{D_i\}_{i=1}^K$ are set on the corresponding clients, respectively. Additionally, we introduce local classifiers $\{C_i\}_{i=1}^K$, one for each client. These classifiers are respectively trained using local data and are aimed to discriminate the semantic consistency between the predictions of the classifier and the preset labels. In the federated enhancement stage, the local classifiers are initialized by the ones trained in the generative adversarial stage and then aggregated into a global model in the server. As shown in Figure 2, the global model is refined with the samples synthesized by generator \mathcal{G} and used to update the local model in the clients, which significantly reduces weight divergence during the federated learning. The detailed training procedure of FedMGD will be introduced in the next section.

Algorithm 1 FedMGD: Generative Adversarial Stage.

Input: The number of clients K , local datasets ξ_i .

Output: Generator \mathcal{G} , classifier $\{C_i\}_{i=1}^K$ and discriminator $\{D_i\}_{i=1}^K$.

- 1: Initialize the parameters of generator $\mathcal{G}^{(0)}$, classifier $\{C_i^{(0)}\}_{i=1}^K$ and discriminator $\{D_i^{(0)}\}_{i=1}^K$;
 - 2: **for** $t = 1, \dots, T$ **do**
 - 3: Uniformly sample a set of clients $\mathcal{A}^{(t)} \in [K]$;
 - 4: Set the preset labels by server sampling and synthesize dataset $\xi_{syn}^{(t)}$ by the generator $\mathcal{G}^{(t-1)}$.
 - 5: Send the synthesized dataset $\xi_{syn}^{(t)}$ to a set of clients $\mathcal{A}^{(t)}$;
 - 6: **for** each client $i \in \mathcal{A}^{(t)}$ in parallel **do**
 - 7: Train discriminator $D_i^{(t)}$ in client i using ξ_i and $\xi_{syn}^{(t)}$ by Eq. (7);
 - 8: Train classifier $C_i^{(t)}$ in client i using ξ_i by cross entropy loss;
 - 9: Compute Realistic Score $S_{real}^{(i,t)}$ for client i on $\xi_{syn}^{(t)}$ by Eq. (5);
 - 10: **end for**
 - 11: Select client k according to $\{S_{real}^{(i,t)}\}_{i \in \mathcal{A}}$ by Eq. (6);
 - 12: Update generator $\mathcal{G}^{(t)}$ using the loss computed by client k as Eq. (7).
 - 13: **end for**
-

4. Method

In this section, we elaborate on our proposed method. We proposed the method which consists of two stages, generative adversarial stage and federated enhancement stage. Then we describe the above stages in detail in Section 4.1 and Section 4.2, respectively.

4.1. Generative Adversarial Stage

To prevent the leakage of real data from clients, we split the conventional GAN into two parts: the generator \mathcal{G} on the server side and discriminators $\{D_i\}_{i=1}^K$ on the client side. In this way, the server can access synthesized samples from global data distribution by generator \mathcal{G} but without the leakage of real data from clients. Let p_i denote the data distribution of the i -th client, where $i \in [K]$, and p_g denote the distribution learned by generator \mathcal{G} . The goal of generator \mathcal{G} is to learn the global data distribution p_{data} . In the conventional distributed GAN [34], the distribution of different clients is assumed to follow the same distribution, that is, $p_i = p_j$ for every pair of different clients, where $i \neq j$. Hence, generator \mathcal{G} is trained to approximate the distribution of real data p_{data} by fitting $\frac{1}{K} \sum_{i=1}^K p_i(x)$. However, due to the effect of label distribution skew, the distribution of local clients is significantly different from each other, that is, $p_i \neq p_j$. Therefore, conventional distributed GAN fails to model global data distribution in the federated learning with label distribution skew.

To solve this problem, an approximate solution was proposed in F2U [36], which points out that $p_{max}(x)$ as Eq. (3) can be regarded as the global optimum for p_{data} in distributed GAN.

$$\begin{aligned}
 p_{max}(x) &= \frac{1}{Z} \max_i p_i(x), \\
 Z &= \int_x \max_i p_i(x) dx,
 \end{aligned} \tag{3}$$

where Z is the normalization constant, $p_{max}(x)$ contains all classes across clients, including rare classes that are only available on a few clients. Since it is infeasible to obtain the true distribution of each client $p_i(x)$, we can leverage an alternative way to update the generator \mathcal{G} . That is, for the synthesized sample \hat{x} , the generator \mathcal{G} selects the discriminator with the largest discriminant probability by Eq. (4) to be updated, which is given as follows:

$$D_{max}(\hat{x}) = \max_i D_i(\hat{x}) \tag{4}$$

This training approach allows learning the rare classes and avoids the negative effect of poorly trained model on the clients. However, the unsupervised approach used by F2U ignores the importance of labeling information. In the

unsupervised label distribution skew scenario, the generated data is likely to be concentrated in only a few classes, which is not conducive to modeling the global data distribution, and thus the use of labels to constrain the type of generated data is necessary. In addition, unlabeled data bring new challenges for the subsequent training of downstream models.

To address the issues mentioned above, we propose a modeling method that can describe the global distribution more comprehensively without compromising client privacy. In the proposed method to better guide the generator \mathcal{G} for global modeling, we set the discriminator \mathcal{D}_i and the classifier \mathcal{C}_i on the client side to constrain the generated data in terms of both semantic truth and image truth, respectively. For the synthetic samples whose preset labels are consistent with the local labels, \mathcal{D}_i and \mathcal{C}_i give scores in terms of both realism and semantics, which in turn guide the generator \mathcal{G} for the next update. The communication process between the server and the client in the FedMGD generative adversarial stage is shown in Figure 1. Specifically, in the communication round t , the generator $\mathcal{G}^{(t)}$ synthesizes dataset $\xi_{syn}^{(t)}$ and feeds them to the randomly selected clients. The synthetic dataset $\xi_{syn}^{(t)}$ is a batch of samples \hat{x} generated according to the specified label \hat{y} . Discriminator $\mathcal{D}_i^{(t)}$ and classifier $\mathcal{C}_i^{(t)}$ in the selected client i output discriminant probability $\mathcal{D}_i^{(t)}(\hat{x})$ and classification probability $\mathcal{C}_i^{(t)}(\hat{x})$, respectively. Then, we further measure the difference between the given label \hat{y} and classification probability $\mathcal{C}_i^{(t)}(\hat{x})$ using cross entropy criterion. Combining discriminant score and classification score, we introduce a novel Realistic Score $S_{real}^{(i,t)}$ for client i at training round t as follows:

$$S_{real}^{(i,t)} = \mathcal{D}_i^{(t)}(\hat{x}) - \mathcal{L}_{xen}(\mathcal{C}_i^{(t)}(\hat{x}), \hat{y}), \quad (5)$$

where $\mathcal{L}_{xen}(\cdot)$ denotes cross entropy metric. The Realistic Score $S_{real}^{(i,t)}$ is used to measure both the realism and semantics of generated images, which provides more comprehensive information for generator \mathcal{G} . We verified the validity of Realistic Score in Section 5.3.1.

To model the global distribution of data across clients with label distribution skew, we follow and improve the solution of F2U. Specifically, we select the discriminator according to Realistic Score $S_{real}^{(i,t)}$, instead of simple discriminant probability. The selected client k , which is trained with generator $\mathcal{G}^{(t)}$, can be denoted as:

$$k = \arg \max_{i \in \mathcal{A}^{(t)}} S_{real}^{(i,t)}, \quad (6)$$

where $\mathcal{A}^{(t)}$ denotes the set of clients selected by the server in the current training round.

Finally, the objective function of the proposed GAN can be presented as:

$$\begin{aligned} \min_{\mathcal{G}^{(t)}} \max_{\mathcal{D}_k^{(t)}} V(\mathcal{D}_k^{(t)}, \mathcal{G}^{(t)}) = & \mathbb{E}_{x \sim p_{data}(x)} [\log \mathcal{D}_k^{(t)}(x)] + \\ & \mathbb{E}_{z \sim p_g(z)} [\log(1 - \mathcal{D}_k^{(t)}(\mathcal{G}^{(t)}(z|\hat{y})))] + \mathcal{L}_{xen}(\hat{y}, \mathcal{C}_k^{(t)}(\mathcal{G}^{(t)}(z|\hat{y}))), \end{aligned} \quad (7)$$

where z is random Gaussian noise. The overall algorithm for generative adversarial stage is shown in Algorithm 1.

In this way, we can obtain additional global data without violating client privacy for federated learning, which can further improves the performance of the global model in the scenario of label distribution skew. In addition, we adopt a conditional generator in our GAN, which synthesizes samples by the given labels in a controllable manner. Specifically, the training process ensures that the data for each class is fully trained by preset labels for the generator \mathcal{G} . In this case, the preset labels are sampled using Server Sampling, i.e., uniform sampling among all classes at the server side. We will discuss the proposed sampling method in Section 5.3.4. Finally, the classifier from the generative adversarial stage can also be directly used in the subsequent federated enhancement stage.

4.2. Federated Enhancement Stage

In federated enhancement stage, we exploit the global generator \mathcal{G} trained in generative adversarial stage to alleviate the performance degradation caused by label distribution skew issue. By modeling the global distribution, the samples synthesized by generator \mathcal{G} can effectively reduce the differences in label distribution among clients. Specifically, the training procedure of this stage can be introduced as follows:

1. We initialize the classifiers $\{\mathcal{C}_i\}_{i=1}^K$ of clients with the ones trained in generative adversarial stage.

2. In the communication round t , we uniformly sample a subset of clients $\mathcal{A}^{(t)} \subseteq [K]$ and train their classifiers using local data.
3. We upload the parameters of classifiers $\{C_i^{(t)}\}_{i \in \mathcal{A}^{(t)}}$ to server and further aggregate the uploaded parameters into the global model $\tilde{C}^{(t)}$.
4. Generator \mathcal{G} synthesizes samples, which follow the global distribution of data across clients, to refine the aggregated global model $\tilde{C}^{(t)}$. Specifically, in this step, we only select generated samples, whose labels are consistent with the predictions of global model $\tilde{C}^{(t)}$, to train global model $\tilde{C}^{(t)}$.
5. The trained global model $\tilde{C}^{(t)}$ is downloaded to the corresponding clients in $\mathcal{A}^{(t)}$ and the training in the next round continues.

The overall algorithm is summarized into Algorithm 2. In this stage, we use the knowledge obtained from global modeling to refine the aggregation model and mitigate the hazards caused by the label distribution skew problem. In this way, we re-correct the problem that the aggregation model is inconsistent with the global optimization direction due to local distribution differences. And because the generator provides additional globally distributed data to the aggregation model, it further improves the performance and generalization of the aggregation model. We prove the validity of the proposed method in Section 5.2.

Algorithm 2 FedMGD: Federated Enhancement Stage.

Input: The number of clients K , generator \mathcal{G} , local datasets $\cup_{i=1}^K \xi_i$, initial classifier parameters $\{\hat{\omega}_i\}_{i=1}^K$.

Output: The global classifier $\tilde{C}^{(t)}$.

- 1: **for** $i = 1$ to K **do**
- 2: Initialize classifier parameters by $\omega_i^{(0)} \leftarrow \hat{\omega}_i$;
- 3: **end for**
- 4: **for** $t = 1$ to T **do**
- 5: Uniformly sample a subset of clients $\mathcal{A}^{(t)} \subseteq [K]$;
- 6: **for** each client $i \in \mathcal{A}^{(t)}$ in parallel **do**
- 7: Train classifier $C_i^{(t)}$ in client i for E epochs;
- 8: Send the parameters $\omega_i^{(t)}$ of classifier $C_i^{(t)}$ to the server;
- 9: **end for**
- 10: Aggregate client model weights $\omega_i^{(t)}$ by:

$$\omega^{(t)} = \sum_{i=1}^K \frac{N_i}{N} \omega_i^{(t)}$$

where N is the number of all data across clients, N_i is the number of data for client i , and K is the number of clients;

- 11: Set the preset labels by server sampling and synthesize dataset $\xi_{syn}^{(t)}$ by the generator \mathcal{G} .
 - 12: Select the consistent synthesize samples $\xi_{syn}^{(t)}$ to refine the global classifier $\tilde{C}^{(t)}$;
 - 13: Download the parameters $\omega^{(t)}$ to update the ones in clients $[K]$.
 - 14: **end for**
-

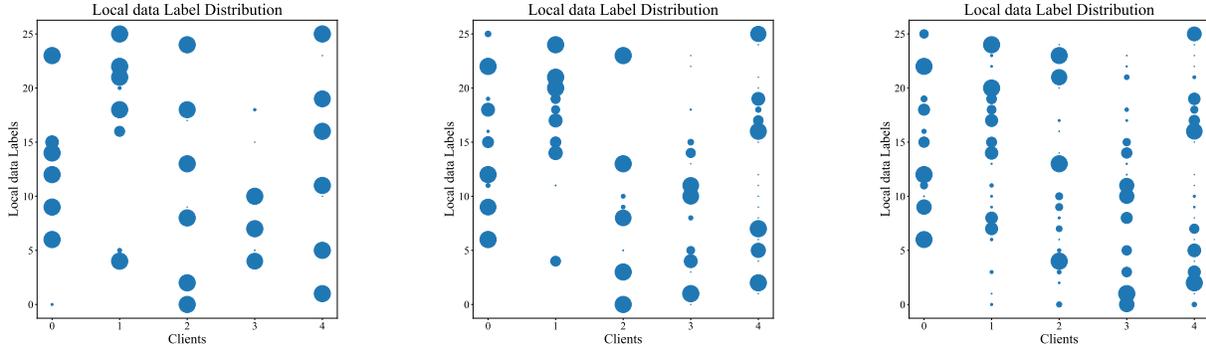
5. Experiments

In this section, we experimentally verify the effectiveness of FedMGD, and summarize the implementation details in Section 5.1. We compare FedMGD with several baseline algorithms in Section 5.2 and further analyze FedMGD by ablation experiments in Section 5.3.

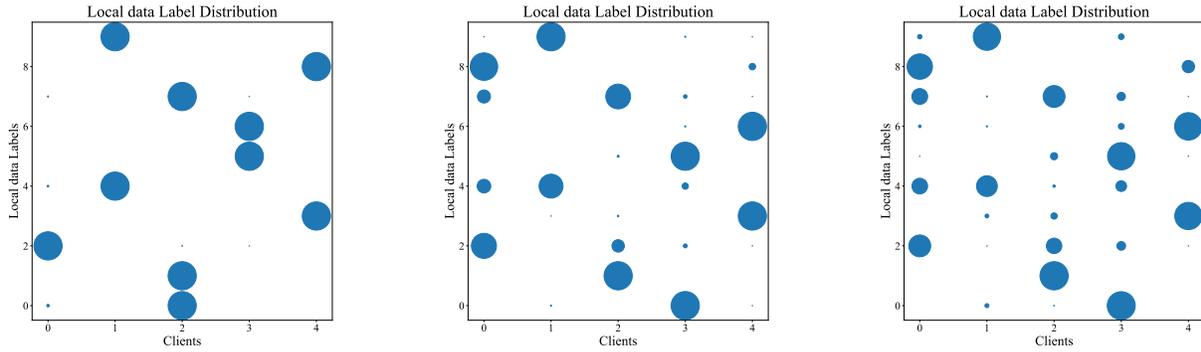
5.1. Experimental Setup

5.1.1. Dataset

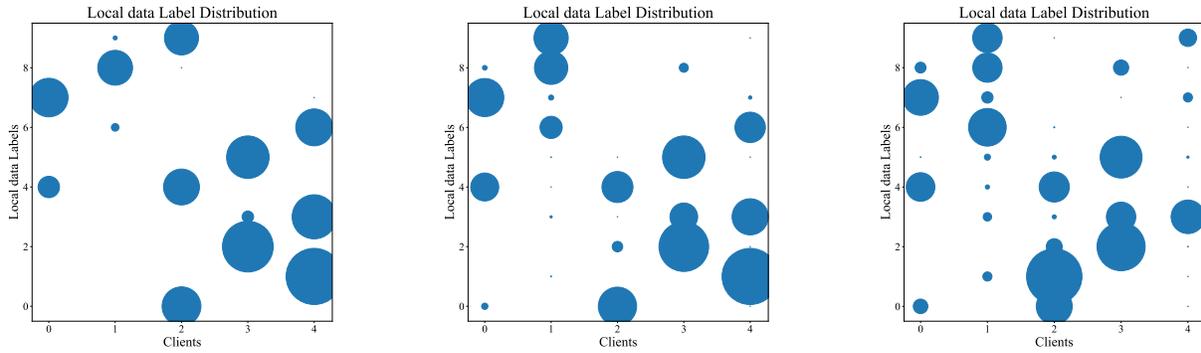
We conduct experiments on the following image datasets: EMNIST [37], FashionMNIST [38], SVHN [39] and CIFAR10 [40]. Among them, EMNIST includes 26 handwritten letters with different kinds of labels and FashionMNIST contains 10 different kinds of clothing images. SVHN is a dataset consisting of different house numbers in street



(a) The EMNIST dataset is divided under 5 clients according to the Dirichlet distribution.



(b) The FashionMNIST dataset is divided under 5 clients according to the Dirichlet distribution.



(c) The SVHN dataset is divided under 5 clients according to the Dirichlet distribution.

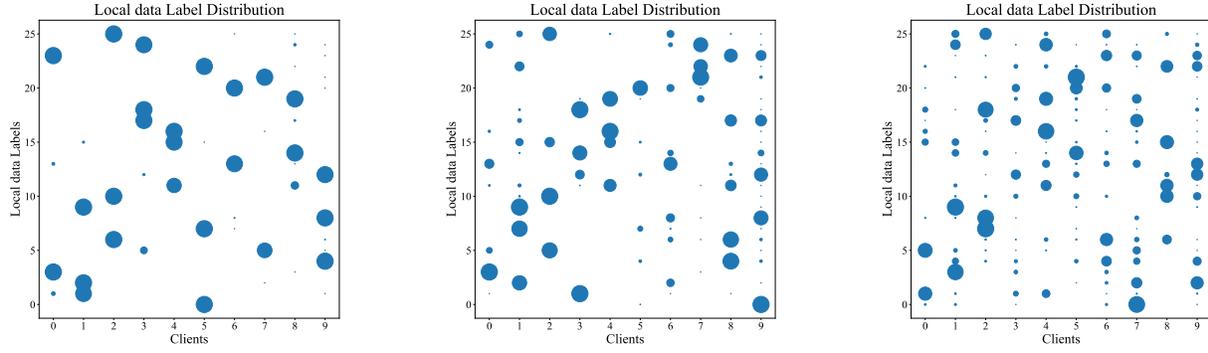
Figure 3: The specific distribution of client labels in different datasets with 5 clients and label distribution skewness (α). Where, the circle represents that the client contains that class of data, and the circle size represents the proportion of that class of labels in the total data of all clients. From left to right α is 0.01, 0.05, and 0.1.

view images. CIFAR10 is a dataset including 10 classes of color images from the real world, making the task more difficult because it contains more noise. We divide 10% of the data as public test set and distribute the rest data over the clients for locally training and testing. The whole process is controlled by random seeds. To ensure the consistency of image resolution, we resize all images to 32×32 .

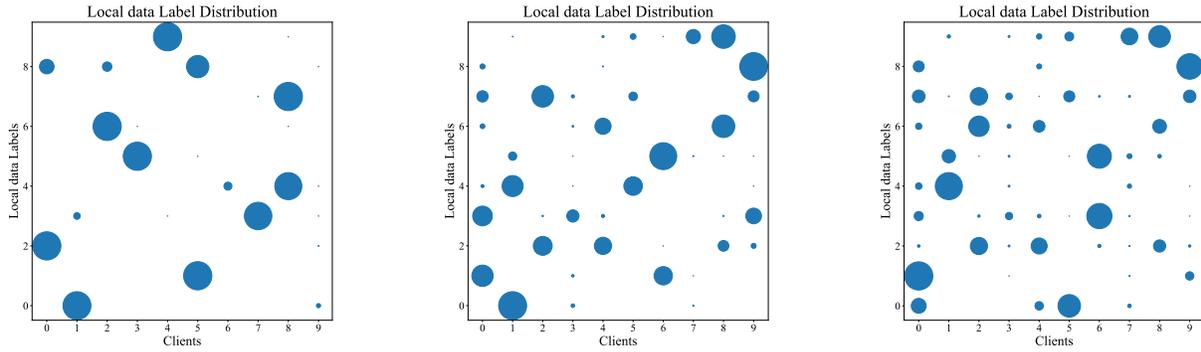
5.1.2. Distribution of Labels among Clients

Following the existing works [41, 9, 29], we use Dirichlet distribution $\text{Dir}(\alpha)$ to simulate the data distribution among clients in the scenario of label distribution skew, where the value of α controls the degree of label distribution

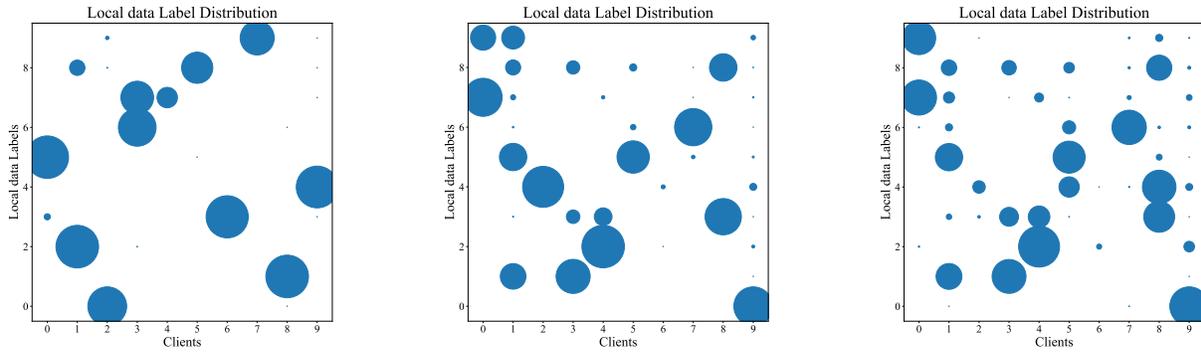
Modeling Global Distribution for Federated Learning with Label Distribution Skew



(a) The EMNIST dataset is divided under 10 clients according to the Dirichlet distribution.



(b) The FashionMNIST dataset is divided under 10 clients according to the Dirichlet distribution.



(c) The SVHN dataset is divided under 10 clients according to the Dirichlet distribution.

Figure 4: The specific distribution of client labels in different datasets with 10 clients and label distribution skewness (α). Where, the circle represents that the client contains that class of data, and the circle size represents the proportion of that class of labels in the total data of all clients. From left to right α is 0.01, 0.05, and 0.1.

skew. The larger the value of α means the smaller the difference in label distribution among clients. Specifically, we set α to 0.1, 0.05, and 0.01 to compare the effect of α on algorithms.

In Figure 3, we show the specific distribution of client labels under 5 clients and label distribution skewness (α). The circle represents that the client contains that category of data, and the circle size represents the proportion of that class of labels in the total data of all clients. In addition, we present in Figure 4 the division of the different datasets according to the Dirichlet distribution for 10 clients.

Table 1

Accuracy comparison on the global test set. FedMGD and state-of-the-art methods train the same model on scenarios with different data sets and data distributions and test the accuracy of the model using a global test set (%).

Dataset	Client Num	α	Local	FedAvg	FedProx	FedDF	FedGen	SCAFFOLD	FedMGD
EMNIST	5	0.01	24.36±0.23	86.56±0.95	85.43±0.61	88.06±0.37	82.41±2.34	85.30±0.37	89.00±0.93 (↑2.44)
		0.05	33.20±0.29	89.33±0.16	87.97±0.40	89.27±0.27	86.86±0.89	89.22±0.21	91.15±0.35 (↑1.82)
		0.1	36.86±0.26	90.85±0.31	89.36±0.55	90.32±0.26	90.12±0.63	91.65±0.33	91.52±0.17 (↓0.13)
	10	0.01	13.38±0.08	65.98±3.95	77.09±1.49	65.72±1.33	66.74±8.45	69.23±1.47	85.04±0.81 (↑15.82)
		0.05	19.03±0.03	82.32±0.35	83.23±0.71	83.19±1.27	81.05±1.69	84.06±1.24	84.87±0.29 (↑0.81)
		0.1	32.22±0.02	88.69±0.47	87.68±0.47	88.97±0.32	88.28±0.49	87.88±0.81	87.83±0.33 (↓1.14)
Fashion MNIST	5	0.01	25.71±0.85	68.38±6.69	69.64±2.14	71.02±0.72	55.67±7.13	55.58±1.90	84.04±0.58 (↑13.02)
		0.05	39.54±1.04	83.37±1.84	81.25±1.13	83.67±0.80	79.61±2.59	77.27±0.68	84.87±0.29 (↑3.90)
		0.1	49.15±0.19	88.28±0.89	86.68±0.89	87.53±0.95	84.15±2.21	86.10±1.20	89.29±1.33 (↑1.01)
	10	0.01	20.65±0.85	53.24±3.20	73.03±2.24	57.48±1.91	56.74±3.61	45.42±8.53	74.37±4.44 (↑1.34)
		0.05	31.23±0.21	80.32±3.25	84.30±1.12	79.65±4.39	80.95±1.23	81.70±0.49	85.71±0.45 (↑3.90)
		0.1	41.61±0.73	85.94±1.51	86.76±0.17	83.97±3.62	85.23±2.44	86.50±0.45	86.79±0.90 (↑0.29)
SVHN	5	0.01	18.11±0.09	73.07±0.25	76.70±0.92	72.67±0.79	57.57±1.47	83.05±0.87	84.14±0.91 (↑1.09)
		0.05	26.80±0.38	84.98±0.92	85.60±0.68	84.99±0.69	67.34±0.84	87.66±0.69	88.62±0.39 (↑0.96)
		0.1	29.23±0.44	88.43±0.91	87.91±0.18	88.65±0.22	68.45±4.17	89.83±0.47	90.47±0.37 (↑0.64)
	10	0.01	14.26±0.18	46.39±1.90	61.71±1.58	47.47±2.56	24.41±1.23	53.39±0.99	76.00±0.79 (↑14.29)
		0.05	13.17±0.06	70.62±0.93	75.09±0.52	72.63±2.20	52.02±2.94	76.04±1.27	76.87±1.26 (↑0.83)
		0.1	14.12±0.95	80.09±0.51	78.11±0.17	80.17±0.52	55.55±5.75	79.77±0.44	79.85±0.10 (↓0.32)
CIFAR10	5	0.01	16.69±0.39	46.73±0.90	51.71±1.21	47.25±1.40	26.73±1.19	54.46±0.99	62.61±1.54 (↑8.15)
		0.05	28.72±0.39	61.61±1.36	60.08±3.19	60.27±0.39	41.86±0.47	64.28±1.43	66.52±0.49 (↑2.24)
		0.1	33.00±1.16	65.77±1.77	65.07±0.40	64.58±0.95	46.61±2.88	67.37±1.02	69.31±0.33 (↑1.94)
	10	0.01	15.76±0.04	38.79±4.97	45.98±0.58	37.06±1.26	26.67±1.10	46.09±2.50	55.13±0.72 (↑9.04)
		0.05	24.95±0.87	52.96±0.24	51.68±0.32	52.07±1.97	27.51±1.76	53.01±0.74	58.26±1.31 (↑5.25)
		0.1	35.04±1.54	58.15±0.94	56.36±0.26	57.89±1.00	43.08±0.55	60.04±1.08	60.60±0.62 (↑0.56)

5.1.3. Baselines

We compare FedMGD with the following baselines to evaluate the effectiveness of FedMGD from different perspectives. First, we compare FedMGD with several state-of-the-art federated learning benchmarks: FedAvg [1], FedProx [3], SCAFFOLD [4], FedDF [9], and FedGen [29]. In addition, to clearly demonstrate the performance of our proposed method for global distribution modeling, we also compare a series of existing distributed generative adversarial network methods: F2U [36], MD-GAN [34], and FedGAN [33].

5.1.4. Implementation Details

We implement all the code of FedMGD in PyTorch, where the structure of the generator and multiple discriminators is based on PatchGAN [42] and implement with the 9-blocks of ResNet [43]. For FedGen's generator, we use the structure in the original paper [29] to output the feature representation of samples after the input has passed through a hidden layer. Furthermore, the classifier in all experiments is a standard Convolutional Neural Network (CNN), which consists of two convolutional layers and one fully connected (FC) layer.

For all methods, we set the number of local training epochs $E = 10$, the size of training batch = 32, and are optimized by Adam optimizer with an initial learning rate of 0.0002. For FedMGD, in the generative adversarial stage the server sends generated data of size 64 to the clients for GAN training at each round of communication. In the federated enhancement stage, the generator synthesizes samples of size 2048 per round to refine the global model. In the classification task, we conduct several experiments and choose the accuracy after the last round of averaging as the final result. In addition, some of the curves are smoothed for better presentation of the results. Finally, to ensure the effectiveness of our method on different number of clients, we have verified it on 5 and 10 clients respectively.

Table 2

Fairness comparison of FedMGD and state-of-the-art methods on local test sets. This is measured by the standard deviation of the performance of the different algorithms on the local test set.

Dataset	α	FedAvg	FedProx	FedDF	FedGen	SCAFFOLD	FedMGD
EMNIST	0.01	5.26	7.15	2.96	13.16	7.30	2.78 ($\downarrow 0.81$)
	0.05	3.81	4.53	4.11	5.53	6.50	2.20 ($\downarrow 1.61$)
	0.1	1.69	2.39	1.94	2.05	1.56	2.61 ($\uparrow 1.05$)
Fashion MNIST	0.01	35.36	23.91	35.87	32.93	39.19	11.35 ($\downarrow 12.56$)
	0.05	19.04	18.82	18.11	24.87	28.09	6.18 ($\downarrow 11.93$)
	0.1	9.01	10.75	9.00	10.46	12.00	5.98 ($\downarrow 3.02$)
SVHN	0.01	39.31	21.90	39.12	33.21	13.83	10.01 ($\downarrow 3.82$)
	0.05	8.79	3.21	9.54	8.31	2.41	3.93 ($\uparrow 1.52$)
	0.1	4.55	3.38	3.98	17.43	1.81	1.56 ($\downarrow 0.25$)
CIFAR10	0.01	27.77	23.31	27.48	18.92	18.57	7.54 ($\downarrow 11.03$)
	0.05	22.74	20.35	21.72	20.62	10.95	10.20 ($\downarrow 0.75$)
	0.1	13.71	14.58	14.26	22.31	15.22	7.21 ($\downarrow 6.50$)

5.2. Comparison with State-of-the-art Methods

In this section, we compare the performance of FedMGD and state-of-the-art methods from two main perspectives: global and local. First, from a global perspective, we use accuracy as a metric to verify the performance of FedMGD and SOAT algorithms in the global test set. Then, from a local perspective, we compare the difference in performance between FedMGD and state-of-the-art algorithms in terms of the fairness of the model among clients.

5.2.1. Accuracy on The Global Test Set.

We compare the accuracy of all algorithms in the global test set under different degrees of label distribution skew, and the results are shown in Table 1. We observe that FedMGD outperforms other state-of-the-art methods in most cases, especially in highly heterogeneous scenario. Figure 5 visualizes the performance of all algorithms under different degrees of label distribution skew. The figure shows that FedMGD exhibits more stable performance when the heterogeneity of the data changes. Compared with other methods, FedGen’s accuracy decreases when there is more noise in training data. This may be because the lightweight generator in FedGen is vulnerable to noise in the sample, resulting in less information about the features captured from the sample. In contrast, FedMGD uses a more sophisticated approach to data generation, thus ensuring the validity of the generated data.

To increase the degree of label distribution skew among the clients, we distribute the labels among 10 clients. Each client has fewer label classes compared to the case with 5 clients. The results are shown in Table 1, where FedMGD can still maintain higher performance compared to other algorithms in the 10 clients scenario. Figure 6 shows how the performance of the different algorithms changes when the data is divided into scenarios with 10 clients. Compared to the scenario with 5 clients, FedMGD has a greater performance improvement compared to the baseline algorithm in the case of a more extreme label distribution. In other words, in scenarios with a more skewed label distribution, the harm caused by the skewed label distribution to the model is reduced because FedMGD uses a global modeling approach to correct the model.

5.2.2. Local Fairness of Different Models.

In the label distribution skew scenario, the same model may have different performance in different clients, and we call this difference in performance between clients the local fairness of the model. We compare the performance of models trained by FedMGD with state-of-the-art Methods on the client test set and demonstrate that the fairness of the models on the client is effectively improved by learning the knowledge of the global distribution. We use the standard deviation of the accuracy of the algorithms on the local test sets of different clients as a measure, and the results are shown in Table 2. We have observed that FedMGD maintains a small difference in performance across clients in all scenarios. This is due to the fact that FedMGD improves the compatibility of the global model under heterogeneous data distribution by using global information to refine the aggregated model, thus allowing the model to perform more fairly (with less performance difference) across clients.

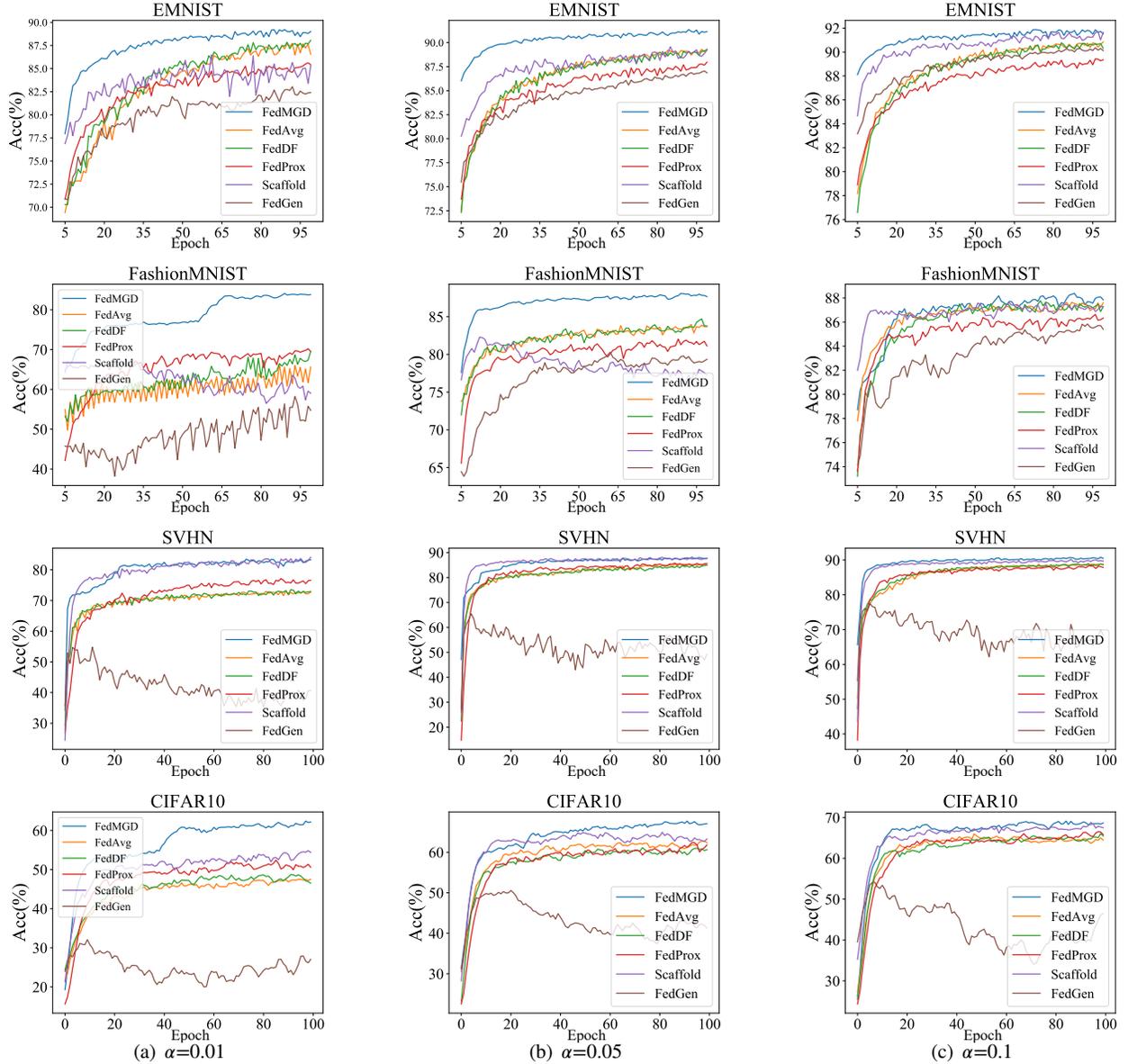


Figure 5: Visualization of algorithm performance at different degrees of label distribution skew. The case where the number of clients is 5 is shown here.

5.3. Ablation Experiment

In this section, we mainly demonstrate the rationality of the proposed FedMGD through several ablation experiments. First, we verify that the proposed Realistic Score is more suitable for the case of label distribution skewing by comparing it with the collected data and F2U. Then, to demonstrate the quality of the data generated by FedMGD in the case of skewed label distribution, we compare it with several distributed generation models. Finally, we validate the rationality of FedMGD's preset label sampling approach.

5.3.1. Realistic Score in FedMGD

In this subsection, we demonstrate from different perspectives that the proposed Realistic Score (Eq. (5)) is beneficial for modeling the global data distribution. Before that, we need a downstream task that can use different data sources and use the model accuracy of the downstream task as a measure. We choose to experiment on a federated distillation algorithm (FedDF [18]) that requires an additional sources, and we can choose different data sources for the additional distillation dataset.

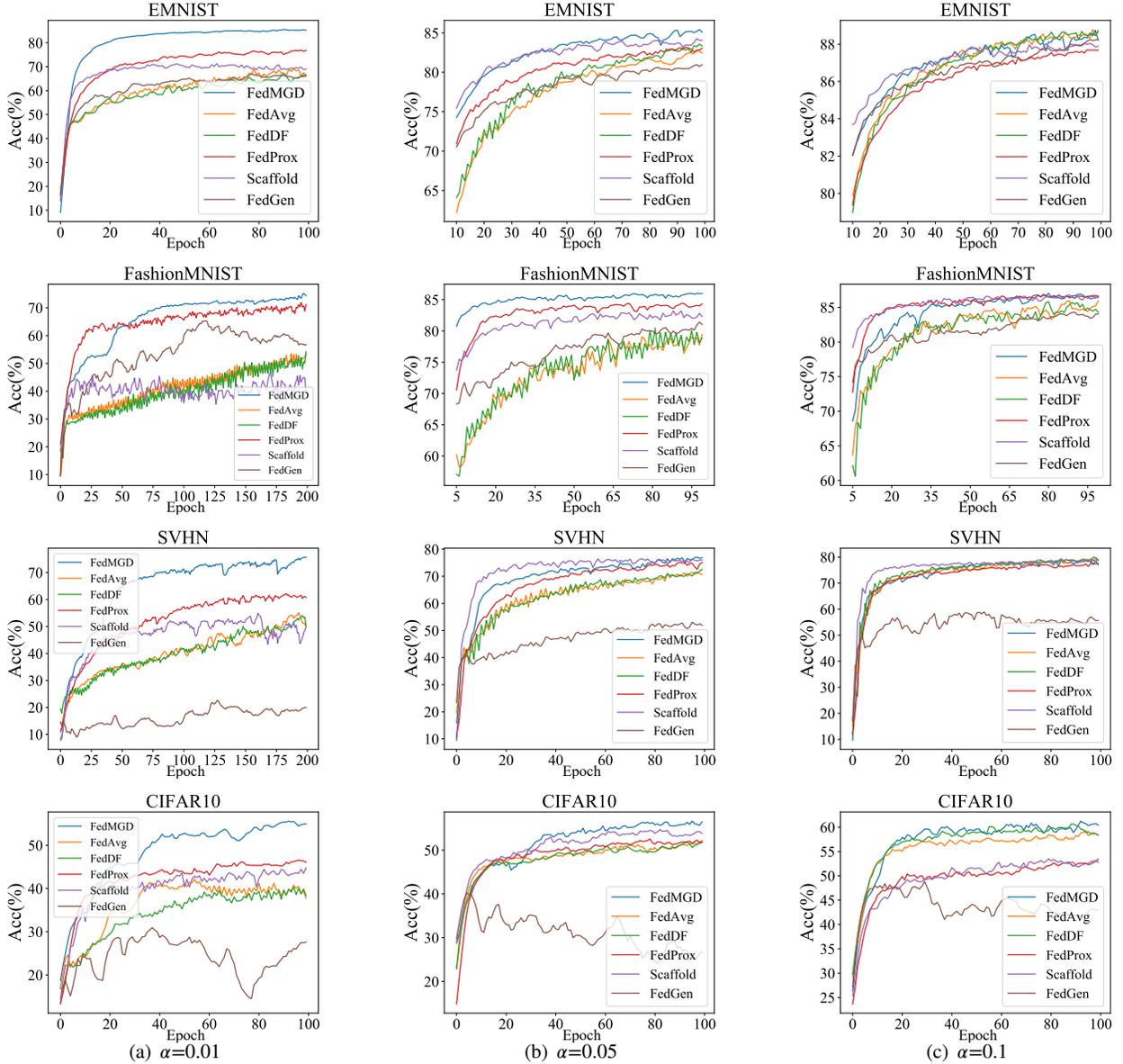


Figure 6: Visualization of algorithm performance at different degrees of label distribution skew. The case where the number of clients is 10 is shown here.

(1) The effect of Realistic Score. We verify the impact of label information of data on model performance when using real data sources as additional data sources. We use 5% real data collected from customers as an additional data source and compare it under different label bias scenarios using unlabeled and labeled methods. As shown in Table 3, the model has better performance when the labeled real data is used as an additional data source, but using the way the data is collected may leak more users' privacy. As shown in Figure 7, FedDF(labeled real data) and FedDF(unlabeled real data) are the accuracies of training models using 5% of real data with and without labels, respectively. The figure demonstrates that the accuracy of the trained model using real labeled data is significantly higher than that of the trained model using unlabeled real data. This proves that labels play an important role in model training and the necessity of introducing semantic truth in the Realistic Score.

(2) The effect of generated data. In this experiment, we use the generator of FedMGD trained by Realistic Score as the data source of FedDF and compare it with the collected real data with labels. The experimental results are shown in Table 5. The generated data using the generator trained by Realistic Score as an additional data source can approximate the effect of real data. We show the accuracy variation of the two approaches in Figure 7, where FedDF(labeled real

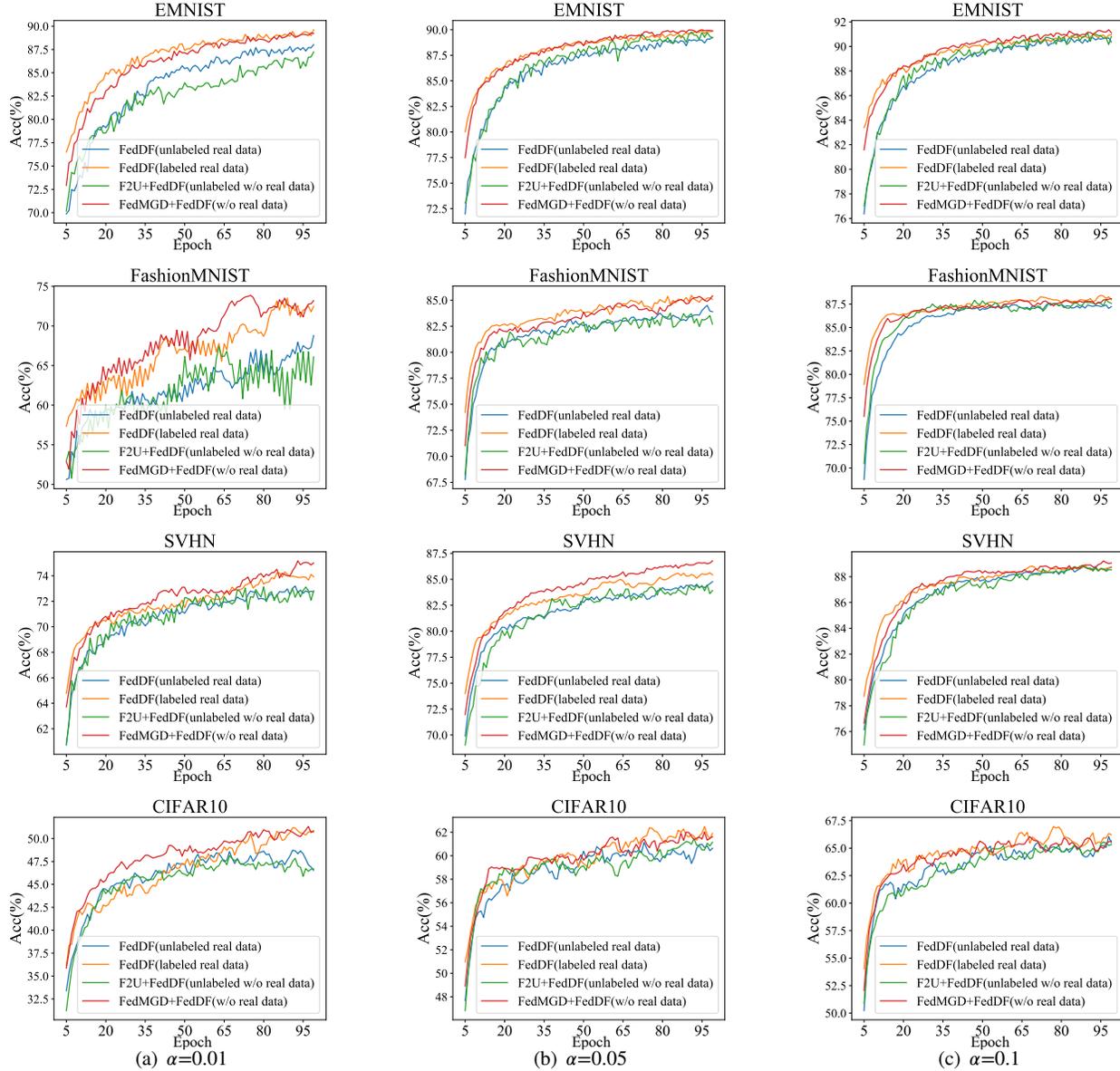


Figure 7: Visualization of the federated distillation performance of different data sources in the label distribution skew scenarios.

data) is the accuracy of training the model using 5% of the real data with labels, and FedMGD+FedDF(w/o real data) is the accuracy of training the model using the generated data after modeling with FedMGD. The figure shows that FedMGD can approximate an accuracy similar to that of using real labeled data by global modeling.

(3) Comparison with F2U. In this experiment, we compare the proposed Realistic Score with the generator trained by the Largest Score proposed in F2U. As shown in Table 4, the generator of FedMGD trained by Realistic Score can reach better results as the data source of FedDF. Because only the data realness is concerned in F2U and ignore the label realness information, the Realistic Score approach is more suitable for global modeling of data under label distribution skew. As shown in Figure 7, F2U+FedDF(unlabeled w/o real data) and FedMGD+FedDF(w/o real data) are the accuracy curves of the trained models using F2U and FedMGD as the data source of FedDF, respectively. Since FedMGD learns more information about the label distribution using Realistic Score, the accuracy of the training model is higher than that of the F2U method.

Table 3

Verify the importance of label information. We used 5% of the real data collected from the client as an additional data source and compared it under different label distribution skew scenarios using both unlabeled and labeled methods.

Dataset	Method	$\alpha=0.01$	$\alpha=0.05$	$\alpha=0.1$
EMNIST	FedDF(unlabeled real data)	87.63±0.30	89.27±0.27	90.32±0.26
	FedDF(labeled real data)	89.69±0.49 (↑2.06)	90.91±0.17 (↑1.64)	90.97±0.30 (↑0.65)
Fashion MNIST	FedDF(unlabeled real data)	71.02±0.72	83.67±0.80	87.53±0.95
	FedDF(labeled real data)	74.03±1.14 (↑3.01)	85.19±1.78 (↑1.52)	88.61±0.44 (↑1.08)
SVHN	FedDF(unlabeled real data)	72.67±0.79	84.99±0.69	88.65±0.22
	FedDF(labeled real data)	73.74±0.84 (↑1.07)	85.23±0.51 (↑0.24)	88.88±0.19 (↑0.23)
CIFAR10	FedDF(unlabeled real data)	47.25±1.40	60.27±0.39	64.58±0.95
	FedDF(labeled real data)	52.86±1.98 (↑5.61)	62.50±0.19 (↑2.23)	66.78±0.23 (↑2.20)

Table 4

Verify the effect of the generated data. The generator(FedMGD) trained by Realistic Score is used as the data source for FedDF and compared with the collected real data with labels (gray areas).

Dataset	Method	$\alpha=0.01$	$\alpha=0.05$	$\alpha=0.1$
EMNIST	FedDF(labeled real data)	89.69±0.49	90.91±0.17	90.97±0.30
	FedMGD+FedDF(w/o real data)	89.19±0.10	89.89±0.41	91.09±0.25 (↑0.12)
Fashion MNIST	FedDF(labeled real data)	74.03±1.14	85.19±1.78	88.61±0.44
	FedMGD+FedDF(w/o real data)	73.22±1.21	85.90±0.49 (↑0.71)	88.17±0.39
SVHN	FedDF(labeled real data)	73.74±0.84	85.23±0.51	88.88±0.19
	FedMGD+FedDF(w/o real data)	75.16±1.28 (↑1.42)	86.98±0.80 (↑1.72)	89.09±0.34 (↑0.21)
CIFAR10	FedDF(labeled real data)	52.86±1.98	62.50±0.19	66.78±0.23
	FedMGD+FedDF(w/o real data)	51.12±1.01	61.83±0.98	65.74±0.52

Table 5

Comparison of FedMGD with F2U. The generator obtained after FedMGD and F2U training is used as an additional data source for FedDF, to compare the performance(%) achieved by the two modeling methods on downstream tasks, respectively.

Dataset	Method	$\alpha=0.01$	$\alpha=0.05$	$\alpha=0.1$
EMNIST	F2U+FedDF(unlabeled w/o real data)	87.88±1.04	89.45±0.34	90.71±0.21
	FedMGD+FedDF(w/o real data)	89.19±0.10 (↑1.22)	89.89±0.41 (↑0.44)	91.09±0.25 (↑0.38)
Fashion MNIST	F2U+FedDF(unlabeled w/o real data)	71.76±0.53	81.47±0.82	87.61±0.37
	FedMGD+FedDF(w/o real data)	73.22±1.21 (↑1.46)	85.90±0.49 (↑4.43)	88.17±0.39 (↑0.56)
SVHN	F2U+FedDF(unlabeled w/o real data)	73.40±0.72	84.30±0.53	88.63±0.25
	FedMGD+FedDF(w/o real data)	75.16±1.28 (↑1.76)	86.98±0.80 (↑2.68)	89.09±0.34 (↑0.46)
CIFAR10	F2U+FedDF(unlabeled w/o real data)	45.99±0.32	61.61±0.40	65.74±0.47
	FedMGD+FedDF(w/o real data)	51.12±1.01 (↑5.13)	61.83±0.98 (↑0.22)	65.74±0.52 (↑0.05)

5.3.2. Compare with Other Distributed GANs.

In this subsection, we validate the image quality generated by FedMGD using the Realistic Score approach in scenarios with label distribution skew and the training effect in downstream tasks by comparing with other distributed GANs.

(1) **Quality evaluation of images generated by FedMGD.** We use F2U [36], MD-GAN [34], and FedGAN [33] as baseline algorithms, and use the distance (FID) between the generated data and the real data distribution as a measure of image quality. In Table 6, we compare the FID of images generated by the FedMGD method and the other three

Table 6

Quality evaluation of images generated by FedMGD. Comparison of the FID of the images generated by FedMGD and other distributed GANs under different data distributions.

Dataset	α	FedGAN	MD-GAN	F2U	FedMGD
EMNIST	0.01	198.37	129.59	39.79	22.86 ($\downarrow 16.93$)
	0.05	121.08	128.59	33.63	19.99 ($\downarrow 13.64$)
	0.1	136.83	95.27	33.35	18.69 ($\downarrow 14.66$)
Fashion MNIST	0.01	230.96	152.29	41.53	39.71 ($\downarrow 1.82$)
	0.05	196.19	158.26	47.35	36.49 ($\downarrow 10.86$)
	0.1	215.98	141.51	46.18	40.21 ($\downarrow 5.97$)
SVHN	0.01	245.91	209.26	140.53	126.04 ($\downarrow 14.94$)
	0.05	199.41	206.34	145.69	139.47 ($\downarrow 6.22$)
	0.1	226.36	208.28	145.37	121.01 ($\downarrow 24.36$)
CIFAR10	0.01	264.69	306.03	218.26	206.17 ($\downarrow 12.09$)
	0.05	255.31	281.67	206.02	201.67 ($\downarrow 4.35$)
	0.1	296.09	315.36	206.09	202.17 ($\downarrow 3.92$)

methods. It can be found that FedMGD has obvious advantages in the label distribution skew scenario.

(2) **Verify the effectiveness of the generator for downstream tasks.** To further verify that the trained generator can be used alone for downstream tasks, we train a separate classifier using the trained generator as the data source (F2U is not involved in this comparison since it can only generate unlabeled data). As shown in Figure 8, compared with other methods, classification model training using the data generated by the FedMGD generator can achieve good performance, which proves that the generator trained using FedMGD can be used for downstream tasks.

5.3.3. Two-Stage vs. One-Stage FedMGD

In this subsection, we experimentally validate the rationality of using two-stage of learning for FedMGD. In order to set up comparison experiments, we propose FedMGD using one-stage for training by fusing the federated enhancement stage and the generative adversarial stage of the two-stage. Specifically, we perform the aggregation of the local classifier $\{C_i\}_{i=1}^K$ at the same time as \mathcal{G} is updated by Realistic Score in Algorithm 1. Then, we refine the aggregated model \tilde{C} by the global information currently learned by the generator.

We conducted experiments on two datasets, FashionMNIST and SVHN, and the results are shown in Table 7. We can observe that the accuracy of FedMGD using two-stage for updating is significantly higher on different datasets than the approach that takes one stage for updating. This is due to the fact that in the original generative adversarial stage, the local classifier only uses the client-side local dataset for learning and describes the local label distribution information to the server through Realistic Score. However, in the process of using a stage update, as the classification model is aggregated in the global modeling process, the local model using aggregation cannot accurately describe the local label distribution information of the client. The generator \mathcal{G} biases the modeling of the global label distribution by the Realistic Score returned locally, which eventually exhibits a degradation of the model performance.

Therefore, in FedMGD we adopt a two-stage update method to improve the performance of the federated learning model on the basis of ensuring the accuracy of the global modeling. We show the variation in accuracy for two different procedures for training the model in Figure 9. It is shown from the figure that the accuracy of the model trained using the two-stage procedure is significantly higher than that using the one-stage procedure. Moreover, the difference in the accuracy of the trained models of the two different procedures gradually increases with the increase of the label distribution skew between clients.

5.3.4. Sampling Methods in FedMGD.

In this section we compare the effects of modeling the global data distribution using different generator preset label sampling methods and validate the rationality of using server-side sampling in FedMGD. Based on the characteristics of data distribution in a distributed environment, we propose two different sampling methods, Client Sampling and Server Sampling. Client sampling refers to collecting the distribution of each label in the client, so that the preset labels of the generator are sampled according to the label distribution in the client. Server Sampling means that the

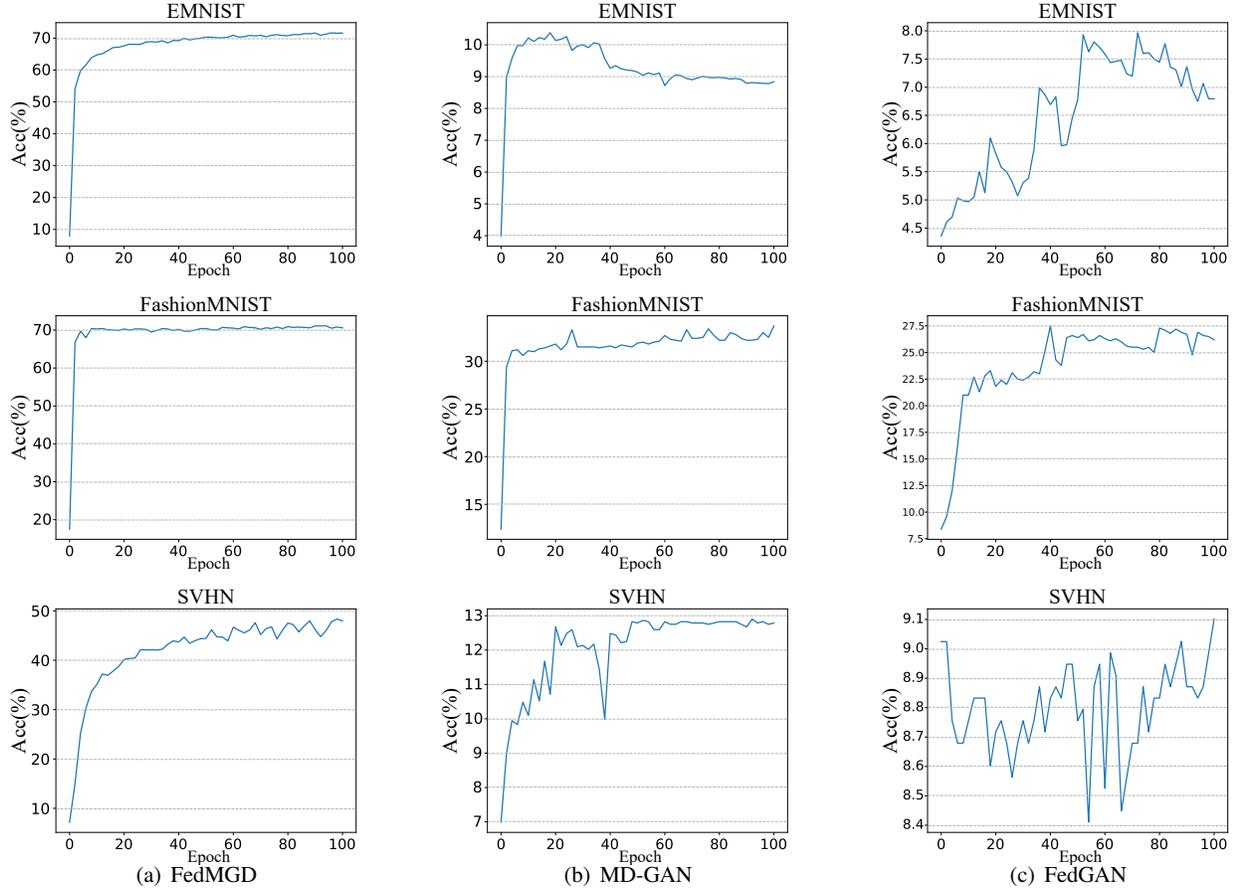


Figure 8: Performance visualization of a separate classifier trained using the generator as data source. At this point α is 0.01.

Table 7

Two-Stage vs. One-Stage FedMGD. FedMGD trains the model in one and two stages to compare the performance of the model under the two procedures (%).

Dataset	Procedure	$\alpha=0.01$	$\alpha=0.05$	$\alpha=0.1$
FashionMNIST	One-stage	51.14 \pm 2.26	80.10 \pm 1.52	85.97 \pm 2.10
	Two-stage	84.04\pm0.58 (\uparrow 32.90)	87.57\pm0.40 (\uparrow 7.47)	89.29\pm1.33 (\uparrow 3.32)
SVHN	One-stage	70.53 \pm 1.41	83.40 \pm 1.56	87.56 \pm 0.25
	Two-stage	84.14\pm0.91 (\uparrow 13.61)	88.62\pm0.39 (\uparrow 5.22)	90.47\pm0.37 (\uparrow 2.91)

generator's preset tags are sampled on the server side using a uniform distribution.

In this experiment, the CIFAR10 data set is used, and the data is divided into 4 clients according to the Dirichlet distribution to ensure that the label distribution and the total amount of data of each client are different. The experimental results are shown in Table 8. In different label distribution skew scenarios, the accuracy of the resulting model is higher when the generator preset label uses the server method to sample the result. This is because the use of uniform sampling in the generative adversarial stage in FedMGD can better learn the data information of each category, which is more conducive to refine the overall label distribution skew. We show the accuracy variation of the two methods in Figure 10. The accuracy of the model trained using server sampling in the figure is significantly higher than client sampling, which demonstrates the validity of our method.

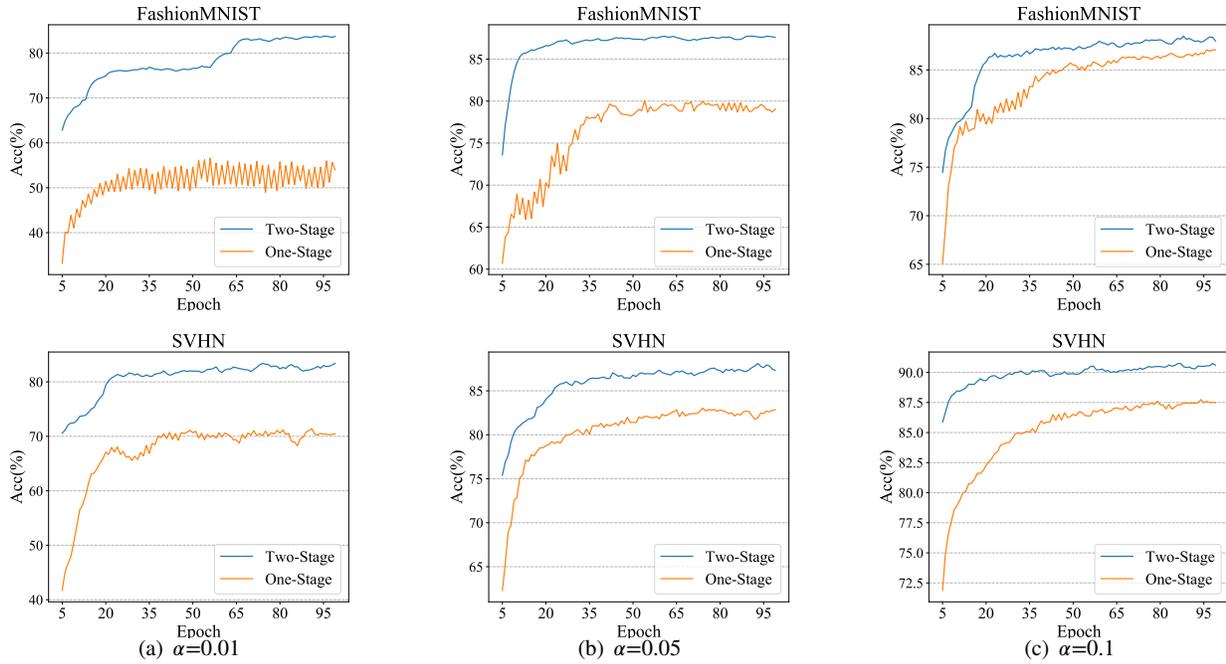


Figure 9: Visualization of two-stage vs. single-stage FedMGD training model performance. The results are obtained when number of clients is 5.

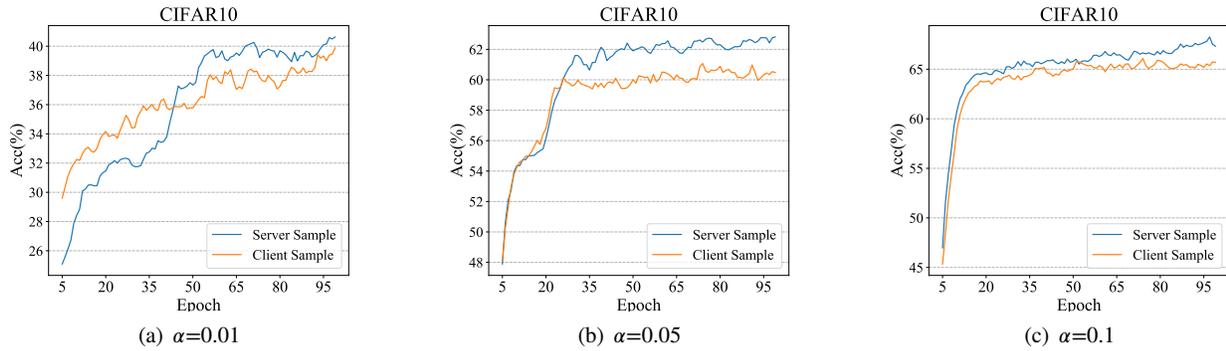


Figure 10: Visualization of algorithm performance under different generator sampling methods. The results are obtained when number of clients is 4.

6. Conclusion

In this work, we mainly investigate how to mitigate the impact of label distribution skew on the performance of federated learning models. We propose a method FedMGD to mitigate the degraded model performance when aggregating heterogeneous data distributions by modeling the global data distribution. It improves the compatibility of the global model under heterogeneous data distribution by using global information to refine the aggregated model. In our experiments, FedMGD has better performance in the scenarios of label distribution skew compared to baseline methods. In addition, we demonstrate through a series of ablation experiments that FedMGD can indeed better model the global data distribution in a label distribution skew scenario and it provides a novel solution for data source of downstream tasks. In future research, we will explore a method for global modeling that is based on any federated learning approach to reduce the performance degradation caused by label distribution skew.

Table 8

Compare the sampling method of FedMGD. Compare the effect of the preset labels obtained by the generator using different sampling methods on the accuracy of the model.

	Client Sample	Server Sample
$\alpha=0.01$	39.46±0.80	40.35±0.87 (↑0.89)
$\alpha=0.05$	60.72±1.58	63.23±0.39 (↑2.51)
$\alpha=0.1$	65.63±0.47	66.91±0.87 (↑1.28)

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (No.2021YFF1201200). This work was carried out in part using computing resources at the High Performance Computing Center of Central South University.

References

- [1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR, 2017, pp. 1273–1282.
- [2] K. Peter, M. H. Brendan, A. Brendan, B. Aurélien, B. Mehdi, B. Arjun, Nitin, et al., Advances and open problems in federated learning, Foundations and Trends in Machine Learning 14 (2021) 1–210.
- [3] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, in: Proceedings of Machine Learning and Systems (MLSys), 2020.
- [4] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A. T. Suresh, Scaffold: Stochastic controlled averaging for federated learning, in: International Conference on Machine Learning (ICML), PMLR, 2020, pp. 5132–5143.
- [5] T. Li, M. Sanjabi, A. Beirami, V. Smith, Fair resource allocation in federated learning, in: International Conference on Learning Representations (ICLR), 2020.
- [6] M. Mohri, G. Sivek, A. T. Suresh, Agnostic federated learning, in: International Conference on Machine Learning (ICML), PMLR, 2019, pp. 4615–4625.
- [7] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated learning with non-iid data, arXiv preprint arXiv:1806.00582 (2018).
- [8] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, R. Yonetani, Hybrid-fl for wireless networks: Cooperative learning mechanism using non-iid data, in: International Conference on Communications (ICC), 2020, pp. 1–7.
- [9] T. Lin, L. Kong, S. U. Stich, M. Jaggi, Ensemble distillation for robust model fusion in federated learning, in: International Conference on Neural Information Processing Systems (NeurIPS), volume 33, 2020, pp. 2351–2363.
- [10] D. Li, J. Wang, Fedmd: Heterogenous federated learning via model distillation, arXiv preprint arXiv:1910.03581 (2019).
- [11] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, S.-L. Kim, Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data, arXiv preprint arXiv:1811.11479 (2018).
- [12] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Transactions on Intelligent Systems and Technology (TIST) 10 (2019) 1–19.
- [13] G. Long, Y. Tan, J. Jiang, C. Zhang, Federated learning for open banking, in: Federated learning, Springer, 2020, pp. 240–254.
- [14] Y. Liu, A. Huang, Y. Luo, H. Huang, Y. Liu, Y. Chen, L. Feng, T. Chen, H. Yu, Q. Yang, Fedvision: An online visual object detection platform powered by federated learning, in: AAAI Conference on Artificial Intelligence (AAAI), volume 34, 2020, pp. 13172–13179.
- [15] W. Li, F. Milletari, D. Xu, N. Rieke, J. Hancox, W. Zhu, M. Baust, Y. Cheng, S. Ourselin, M. J. Cardoso, et al., Privacy-preserving federated brain tumour segmentation, in: International workshop on machine learning in medical imaging (MLMI), Springer, 2019, pp. 133–141.
- [16] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, Journal of Healthcare Informatics Research 5 (2021) 1–19.
- [17] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, W. Shi, Federated learning of predictive models from federated electronic health records, International journal of medical informatics 112 (2018) 59–67.
- [18] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, et al., The future of digital health with federated learning, NPJ digital medicine 3 (2020) 1–7.
- [19] B. Tan, B. Liu, V. Zheng, Q. Yang, A federated recommender system for online services, in: ACM Conference on Recommender Systems (RecSys), 2020, pp. 579–581.
- [20] L. Minto, M. Haller, B. Livshits, H. Haddadi, Stronger privacy for federated collaborative filtering with implicit feedback, in: ACM Conference on Recommender Systems (RecSys), 2021, pp. 342–350.
- [21] J. Wu, Q. Liu, Z. Huang, Y. Ning, H. Wang, E. Chen, J. Yi, B. Zhou, Hierarchical personalized federated learning for user modeling, in: The Web Conference (WWW), 2021, pp. 957–968.
- [22] B. Hitaj, G. Ateniese, F. Perez-Cruz, Deep models under the gan: Information leakage from collaborative deep learning, in: ACM SIGSAC Conference on Computer and Communications Security (CCS), ACM, 2017, p. 603–618.

- [23] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, Y. Zhou, A hybrid approach to privacy-preserving federated learning, in: ACM workshop on artificial intelligence and security (AISec), 2019, pp. 1–11.
- [24] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, *Future Generation Computer Systems* 115 (2021) 619–640.
- [25] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, J. Passerat-Palmbach, A generic framework for privacy preserving deep learning, arXiv preprint arXiv:1811.04017 (2018).
- [26] X. Li, M. Jiang, X. Zhang, M. Kamp, Q. Dou, Fed{bn}: Federated learning on non-{iid} features via local batch normalization, in: International Conference on Learning Representations (ICLR), 2021.
- [27] J.-H. Duan, W. Li, S. Lu, Feddna: Federated learning with decoupled normalization-layer aggregation for non-iid data, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 722–737.
- [28] Y. Han, X. Zhang, Robust federated learning via collaborative machine teaching, in: AAAI Conference on Artificial Intelligence (AAAI), volume 34, 2020, pp. 4075–4082.
- [29] Z. Zhu, J. Hong, J. Zhou, Data-free knowledge distillation for heterogeneous federated learning, in: International Conference on Machine Learning (ICML), PMLR, 2021, pp. 12878–12889.
- [30] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, B. Courville, et al., Generative adversarial nets, in: International Conference on Neural Information Processing Systems (NeurIPS), volume 27, MIT Press, 2014, pp. 2672–2680.
- [31] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, H. Qi, Beyond inferring class representatives: User-level privacy leakage from federated learning, in: International Conference on Computer Communications (INFOCOM), IEEE, 2019, pp. 2512–2520.
- [32] S. Augenstein, H. B. McMahan, D. Ramage, S. Ramaswamy, P. Kairouz, M. Chen, R. Mathews, et al., Generative models for effective ml on private, decentralized datasets, in: International Conference on Learning Representations (ICLR), 2020.
- [33] M. Rasouli, T. Sun, R. Rajagopal, Fedgan: Federated generative adversarial networks for distributed data, arXiv preprint arXiv:2006.07228 (2020).
- [34] C. Hardy, E. Le Merrer, B. Sericola, Md-gan: Multi-discriminator generative adversarial networks for distributed datasets, in: International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2019, pp. 866–877.
- [35] Q. Chang, H. Qu, Y. Zhang, M. Sabuncu, C. Chen, T. Zhang, D. N. Metaxas, Synthetic learning: Learn from distributed asynchronous discriminator gan without sharing medical image data, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 13856–13866.
- [36] R. Yonetani, T. Takahashi, A. Hashimoto, Y. Ushiku, Decentralized learning of generative adversarial networks from non-iid data, arXiv preprint arXiv:1905.09684 (2019).
- [37] G. Cohen, S. Afshar, J. Tapson, A. van Schaik, Emnist: Extending mnist to handwritten letters, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 2921–2926.
- [38] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747 (2017).
- [39] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.
- [40] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).
- [41] T.-M. H. Hsu, H. Qi, M. Brown, Measuring the effects of non-identical data distribution for federated visual classification, arXiv preprint arXiv:1909.06335 (2019).
- [42] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1125–1134.
- [43] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778.

Tao Sheng received the B.S. degree in Software Engineering from the Xinjiang University, Urumqi, Xinjiang, China, in 2020. He is currently pursuing the M.S. degree in Software Engineering with the Central South University, Changsha, China. His current research interests include federated learning and deep learning.

Chenchao Shen received the PhD degree from Zhejiang University, China, in 2021. He is currently an associate professor at the School of Computer, Central South University, Changsha, Hunan, China. His research interests include computer vision, deep learning, and self-supervised learning.

Yuan Liu obtained his Bachelor’s degree in Software Engineering and is currently a Ph.D. student in the School of Computer Science and Engineering at Central South University where he works on federated learning for different distributions and generalization.

Yeyu Ou received the B.S. degree in Engineering Mechanics from the Hehai University, Nanjing, China, in 2020. She is currently pursuing the M.S. degree in Computer Technology with the Central South University, Changsha, China. Her current research interests include graph neural networks and big data analysis.

Zhe Qu received the M.S. degree from the University of Delaware, Newark, DE, USA, in 2017. He is currently

a Ph.D. candidate in the University of South Florida, Tampa, FL, USA. His research interests include network and mobile system security, and machine learning for networks.

Jianxin Wang (Senior Member, IEEE) received the Ph.D. degree from Central South University, Changsha, Hunan, China. He is currently the Chair and a Professor with the School of Computer Science and Engineering, Central South University. His research interests include algorithm analysis and optimization, parameterized algorithm, bioinformatics, and computer networks.

Appendix

arXiv:2212.08883v1 [cs.LG] 17 Dec 2022

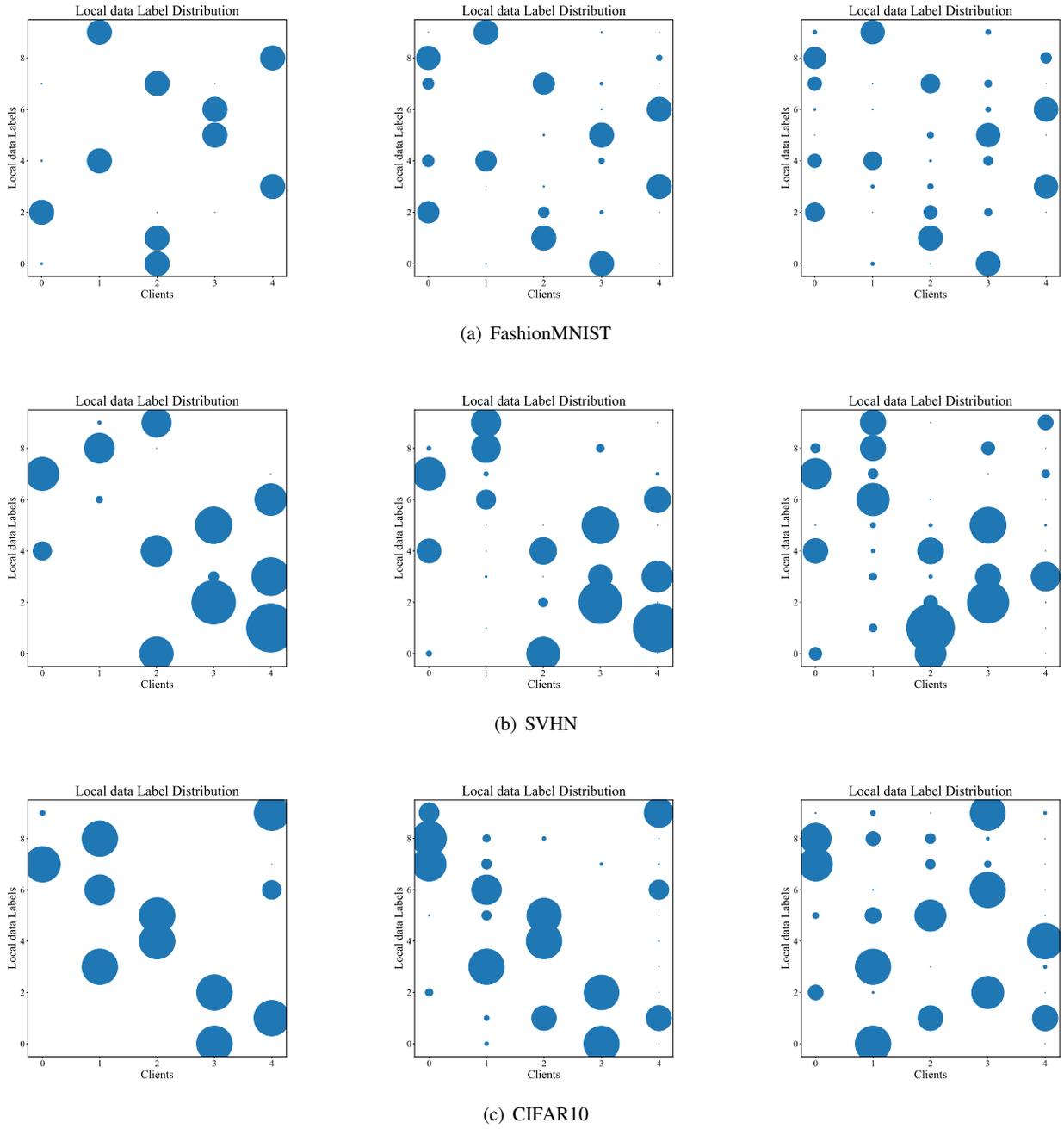


Figure 1: Visualization of statistical heterogeneity between users on different datasets when the number of clients is 5. Where the x-axis represents the different clients, the y-axis indicates the class labels, and the size of the scattered points indicates the number of training samples with available labels for that user. From left to right α is 0.01, 0.05, and 0.1.

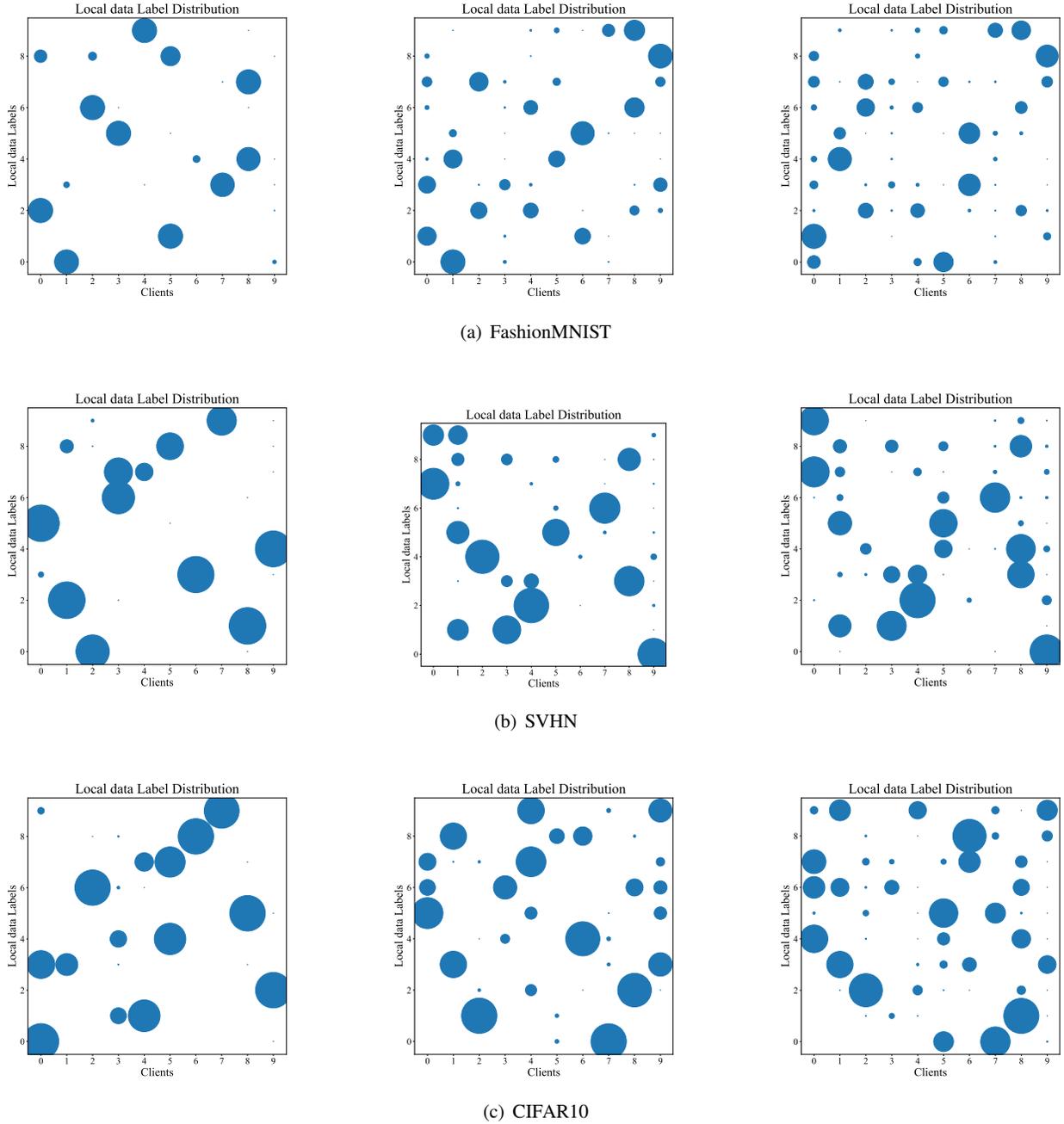


Figure 2: Visualization of statistical heterogeneity between users on different datasets when the number of clients is 10. Where the x-axis represents the different clients, the y-axis indicates the class labels, and the size of the scattered points indicates the number of training samples with available labels for that user. From left to right α is 0.01, 0.05, and 0.1.

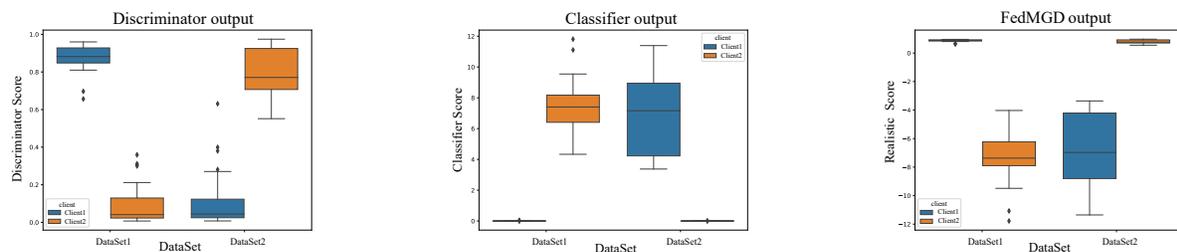


Figure 3: The classifiers and discriminators are trained in two clients with different data classes. To verify that the classifiers and discriminators perform worse on data categories that have never been seen before, we take each other's data as input. From left to right, the discriminator score, the classifier score (distance from the true label), and the FedMGD (merged) realistic score are shown.