# Data-Free Knowledge Distillation via Generator-Free Data Generation for Non-IID Federated Learning

**Siran Zhao**
Sun Yat-sen University
**Tianchi Liao**
Sun Yat-sen University
**Lele Fu**
Sun Yat-sen University
**Chuan Chen** ( chenchuan@mail.sysu.edu.cn )
Sun Yat-sen University
**Jing Bian**
Sun Yat-sen University
**Zibin Zheng**
Sun Yat-sen University

**Research Article**

**Keywords:** Non-IID Federated Learning, Data Heterogeneity, Data Generation, Data-Free Knowledge Distillation

**Posted Date:** September 26th, 2023

**DOI:** https://doi.org/10.21203/rs.3.rs-3364332/v1



**Additional Declarations:** No competing interests reported.

# Data-Free Knowledge Distillation via Generator-Free Data Generation for Non-IID Federated Learning

Siran Zhao[1], Tianchi Liao[2], Lele Fu[3], Chuan Chen[1*], Jing Bian[1], Zibin Zheng[2]

[1*]Sun Yat-sen University, School of computer science and engineering, Guangzhou, China.
[2*]Sun Yat-sen University, School of software engineering, Zhuhai, China.
[3*]Sun Yat-sen University, School of systems science and engineering, Guangzhou, China.

*Corresponding author(s). E-mail(s): chenchuan@mail.sysu.edu.cn;
Contributing authors: zhaosr3@mail2.sysu.edu.cn;
liaotch@mail2.sysu.edu.cn; fulle@mail2.sysu.edu.cn;
mcsbj@mail.sysu.edu.cn; zhzibin@mail.sysu.edu.cn;

**Abstract**

Data heterogeneity (Non-IID) on Federated Learning (FL) is currently a widely publicized problem, which leads to local model drift and performance degradation. Because of the advantage of knowledge distillation, it has been explored in some recent work to refine global models. However, these approaches rely on a proxy dataset or a data generator. First, in many FL scenarios, proxy dataset do not necessarily exist on the server. Second, the quality of data generated by the generator is unstable and the generator depends on the computing resources of the server. In this work, we propose a novel data-Free knowledge distillation approach via generator-Free Data Generation for Non-IID FL, dubbed as FedF$^2$DG. Specifically, FedF$^2$DG requires only local models to generate pseudo datasets for each client, and can generate hard samples by adding an additional regularization term that exploit disagreements between local model and global model. Meanwhile, FedF$^2$DG enables flexible utilization of computational resources by generating pseudo dataset locally or on the server. And to address the label distribution shift in Non-IID FL, we propose a Data Generation Principle that can adaptively control the label distribution and number of pseudo dataset based on client current state, and this allows for the extraction of more client knowledge. Then

knowledge distillation is performed to transfer the knowledge in local models to the global model. Extensive experiments demonstrate that our proposed method significantly outperforms the state-of-the-art FL methods and can serve as plugin for existing Federated Learning methds such as FedAvg, FedProx, etc, and improve their performance.

**Keywords:** Non-IID Federated Learning, Data Heterogeneity, Data Generation, Data-Free Knowledge Distillation

# 1 Introduction

With the explosive growhth of data and the growing emphaises on privacy protection, the conventional AI methods that require uploading source data to a central server for training have become untrustworthy. Recently, Federated Learning (FL) [1] has been proposed to address the privacy security risks in conventional AI. It allows multiple devices to train a shared global model without uploading the local source data to a central server, which is an effective way to leverage data from multiple devices and protect user privacy [2, 3]. FL has been successfully applied in real-world applications, such as health care [3–5] and recommender system [6, 7], etc.

The main problem that FL currently faces is data heterogeneity, i.e., the client data in real scenarios is usually non-identically and independently distributed (Non-IID). Under the Non-IID setting, the vanilla FL algorithm such as FedAvg [1] can encounter the issue of local models drift and forgetting global knowledge catastrophically, leading to a decrease in performance [8, 9]. This is because client local training in FL utilizes only local data, i.e., minimizing the local loss. However, the two goals of minimizing the local loss and minimizing the global loss are inconsistent in Non-IID FL [9, 10]. Therefore, the approach of element-wise averaging of local models cannot fully capture the valuable information from clients and may not result in an ideal global model [11].

To address the aforementioned challenge of data heterogeneity, existing works has approached the problem from two main perspectives: One focuses on constraining the direction of local model update to align the local and global optimization objectives [9, 12–15], such as FedProx, SCAFFOLD, etc. In this way, the problem of local model drift can be solved. However, these methods merely do simple model aggregation to obtain the global model, which can result in the cancellation of local knowledge during the aggregation process and prevent the global model from effectively learning the diverse knowledge from different clients [16, 17]. Another focuses on improving the effectiveness of model aggregation, where knowledge distillation has been applied as an effective method [18–22]. Knowledge distillation can improve the global model by extracting knowledge from local models, thereby alleviating the issue of local models drift caused by data heterogeneity. However, these methods rely on an unlabeled dataset as the proxy, which is often impractical in many real-world applications where public datasets may not be always available on the server. Recently, serval data-free knowledge distillation methods for federated learning have been proposed, such as FedGen [23] and FedFTG [24]. They both apply a generator [25] to generate pseudo

**Table 1**: Advantages of FedF$^2$DG over other methods

| | Align the local and global optimization objectives | Boost model aggregation | No proxy dataset is required | Flexible use of computational resources |
|---|---|---|---|---|
| FedAvg | | | | |
| FedProx, SCAFFOLD, etc | ✔ | | | |
| data-dependent knowledge distillation methods for FL | | ✔ | | |
| data-free knowledge distillation methods for FL | | ✔ | ✔ | |
| FedF$^2$DG | ✔ | ✔ | ✔ | ✔ |

data. Nevertheless, the generator requires iterative training [26] and can only be stored on the server, these methods face issues of unstable generated data quality and great dependence on the server's computational resources. In Table 1, we summarize the advantages of our approach FedF$^2$DG over existing works. Compared to methods that constrain the direction of local model update such as FedProx, we add the step of boosting model aggregation through knowledge distillation. Compared to data-dependent knowledge distillation methods for FL, we do not need an additional proxy dataset but rather generate pseudo datasets through generator-free data generation. Compared to data-free knowledge distillation methods for FL, we do not need to train the data generator and have the flexibility to utilize the computational resources of the client and server to generate data. Moreover, compared with the knowledge distillation methods for FL, our method can be combined with methods for optimizing local model training, thus adding the step of constraining the update direction of local models.

Observing the challenge of data heterogeneity and the limitations of existing works, in this work, we propose a novel **Fed**erated Learning data-**F**ree knowledge distillation approach via generator-**F**ree **D**ata **G**eneration for Non-IID scenarios, called FedF$^2$DG. Specifically, FedF$^2$DG is divided into three Stages. First, in Federated Learning Stage, FedF$^2$DG trains local models with excellent performance in preparation for data generation. Second, in Adaptive Data Generation Stage, to address the issue of requiring data for knowledge distillation, we propose a Data Generation Method that require only local models to generate pseudo datasets for each client by optimizing noise into real images using a regularization term, and can generate hard samples to help the constantly updated global model to learn knowledge not yet learned in clients. Compared with the methods of generating data via the generator, FedF$^2$DG enables flexible utilization of computational resources by generating pseudo dataset locally or on the server. And to address the label distribution shift in Non-IID scenarios, we propose a Data Generation Principle that can adaptively control the label distribution and number of pseudo dataset based on client current state, and this ensures that the global model can focus on learning useful information about each client. Finally, in Knowledge Distillation Stage, knowledge distillation between local models and global model is performed to boost global model performance. The framework of FedF$^2$DG

is shown in Figure 1.

The main contributions in this work are as follows:

- We propose FedF$^2$DG, which enhances the global model aggregation step by enabling data-free knowledge distillation through generator-free data generation.
- We propose a Data Generation Method that leverages only local models by optimizing noise into real images using a regularization term, and can generate hard samples by adding an additional regularization term that exploit disagreements between local model and global model. It enables flexible use of computational resources by generating pseudo dataset locally or on the server.
- We propose a Data Generation Principle that adaptively controls the label distribution and number of pseudo dataset based on client current state. This approach allows for the incorporation of a greater amount of client knowledge into the pseudo dataset.
- We demonstrate that FedF$^2$DG can be combined with methods that optimize local model training, such as FedAvg, FedProx, SCAFFOLD, MOON and FedNova. And can further improve their performance.

# 2 Related Work

## 2.1 Data-Free Knowledge Distillation in Centralized Scenario

Data-Free Knowledge Distillation methods [27–32] are able to generate pseudo data from a pretrained teacher model, and leverage them to transfer the knowledge from teacher model to student model. Lopes *et al.* [29] propose to extract the metadata from the teacher's activation layers and reconstruct the training samples. DAFL [27] and DFAD [28] both train a generator for image generation, where DAFL treats the teacher model as a fixed discriminator, and DFAD employ an adversarial training framework to extract the knowledge from the teacher model. FastDFKD [32] learns a meta-synthesizer that seeks common features as the initialization for the fast data synthesis, enabling fast data synthesis. DeepImpression [30] models the output space of teacher model and update random noise images to obtain training data. DeepInversion [31] synthesizes real images from random noise by regularizing the distribution of intermediate feature maps. However, data-free knowledge distillation in centralized scenario is difficult to apply because of data privacy issues. Therefore, our FedF$^2$DG migrate the data-free knowledge distillation method from the centralized scenario to the federated scenario, which can protect user data privacy.

## 2.2 Federated Learning on Non-IID Data

Federated Learning was first proposed by [1], namely FedAvg, which perfroms weighted aggregation of the local models on the server. A wealth of work has been proposed to address the main challenge in FL: Non-IID. FedProx [9] constrains the local model to be closer to the global model by adding a regularization term on the local loss function. SCAFFOLD [12] uses control variables to keep local updates from drifting. MOON [13] conducts contrastive learning in model-level to correct the drifted local update. FedNova [14] corrects model aggregation scheme to eliminates objective inconsistency.

In summary, the above methods mainly focus on constraining the direction of local model update to align the local and global optimization objectives, while ignoring the loss of useful information during simple aggregation of the global model.

### 2.3 Knowledge Distillation in Federated Learning

Many researches utilize knowledge distillation to address data heterogeneity in FL. And most existing work [18–22] is data-dependent. FedMD [19] enbales participants to independently design their models and to translate knowledge between participants through knowledge distillation. FedDF [18] proposes an ensemble distillation method for model fusion, where global model is trained with unlabeled data generated by local models. FedAUX [20] utilizes an auxiliary dataset for knowledge distillation to implement initialized server models and weighted integrated user models. MHAT [21] utilizes knowledge distillation to extract local models information and trains an auxiliary model for information aggregation. FedAD [22] uses a public dataset for inter-node communication and employs a knowledge distillation algorithm that uses both final prediction and model attention to extract client knowledge. But all these methods rely on a public dataset, which in many FL scenarios do not necessarily exist on the server.

Recently serval works [23, 24, 33, 34] has been proposed to implement data-free knowledge distillation for FL. FedGKD [34] prevents the local model drift by guiding local model training through knowledge distillation between historical global and local models. However, this work focuses on local model updates rather than enhanced model aggregation. FedBKD [33] integrates knowledge distillation into the local model upload and global model download steps of federated learning. FedGen [23] uses a lightweight generator to generate pseudo features, which are then used to help client local updates. FedFTG [24] generates pseudo data by learning a generator and then uses pseudo data to fine-tune the global model. However, FedBKD, FedGen and FedFTG rely on a generator [25] to generate pseudo data. The utilization of generators entails the following challenges: 1) Unstable quality of the generated data. Since the generator needs to be trained iteratively [26], the quality of the generated data may be poor in the initial training phases. 2) Dependence on the server's computational resources. Since the generator can only be stored on the server, the generation of data takes place exclusively on the server, leading to a huge workload burden on the server.

## 3 Proposed Method

In this section, we describe the proposed data-free knowledge distillation method for Non-IID FL: FedF$^2$DG. Considering the limitations of traditional methods using generators to generate pseudo data, we adopt generator-free data generation method named DeepInversion [31], each client can generate pseudo dataset locally or on the server by adaptively utilizing its local model. Meanwhile, due to the different label distributions among clients and the different numbers of data among clients, we define a data generation principle, the label distribution and number of pseudo datasets are adaptively adjusted according to the data generation principle. The global model works with clients-generated pseudo datasets and performs knowledge distillation with their local models to learn the knowledge lost in the simple weighted aggregation process.

Figure 1 shows the framework of our FedF$^2$DG method and the corresponding algorithm is summarized in Algorithms 1. FedF$^2$DG is executed into three stages: 1) Federated Learning Stage. Before the incorporating round $I$, Federated Learning Stage is employed to distribute the global model to clients for local training and to update the global model through simple model aggregation, while after the incorporating round $I$, Federated Learning Stage only distributes global model to clients for local training and the global model is updated at Knowledge Distillation Stage. 2) Adaptive Data Generation Stage, which uses local models to generate pseudo datasets needed for knowledge distillation for the later Knowledge Distillation Stage. 3) Knowledge Distillation Stage, which updates the global model by using pseudo datasets to perform knowledge distillation between the local models and global model.

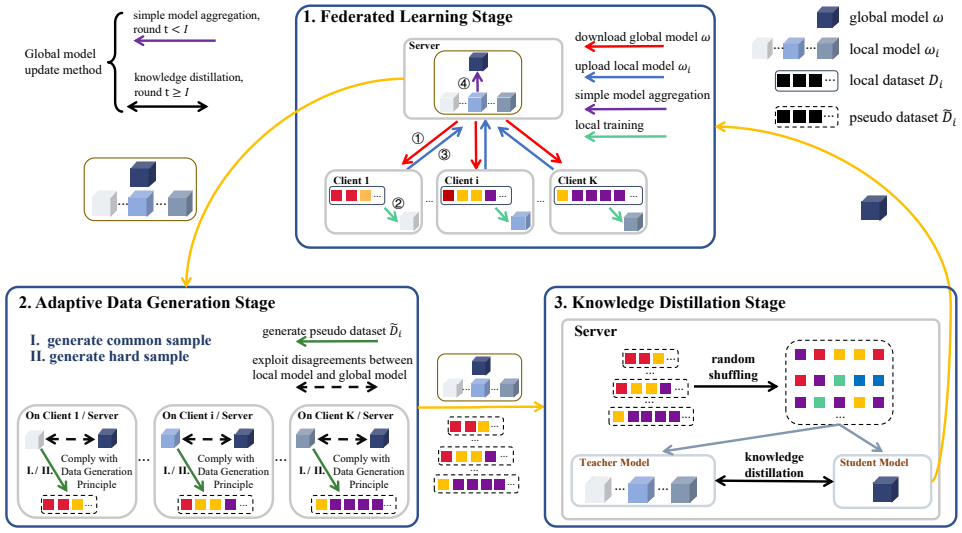



**Fig. 1**: The framework of our FedF$^2$DG method. It's consisted of three stages: 1) Federated Learning Stage, 2) Adaptive Data Generation Stage, 3) Knowledge Distillation Stage.

Our FedF$^2$DG method runs as follows: before the incorporating round $I$, FedF$^2$DG conducts Federated Learning Stage individually, this is because our proposed generator-free data generation method in Adaptive Data Generation Stage requires local models have excellent performance to ensure the quality of the generated pseudo datasets. After the incorporating round $I$, our second and third stage are added. These three stages will then run in a cycle. Note that FedF$^2$DG can be combined with methods that optimize local model training, such as FedAvg, FedProx, SCAFFOLD, MOON and FedNova.

**Algorithm 1** FedF$^2$DG

**Input:** The communication round $T$; The incorporating round $I$; Client number $K$; The fraction of clients selected in each round $F$; Local training epoch $E$; The datasets of clients $\{\mathcal{D}_k\}_{k\in\{1,...,K\}}$; The parameter of the global model $\omega$.

Random initialize the parameter of the global model $\omega$.
**for** $t = 1, ..., T$ **do**
  **1. Federated Learning Stage**
  $S \leftarrow$ (random set of $F \times K$ clients)
  **for** each client $k \in S$ **in parallel do**
    //each client runs E epochs local update.
    $\omega_k^t \leftarrow$ClientUpdate($\omega^{t-1}$,$\mathcal{D}_k$)
  **end for**
  **if** $t \geq I$ **then**
    //incorporating the two stages of Adaptive Data Generation and Knowledge Distillation.
    **2. Adaptive Data Generation Stage**
    compute the number and the label distribution of the pseudo datasets according to Eq. (11) and Eq. (12)
    generate the pseudo datasets $\left\{\tilde{\mathcal{D}}_k^t\right\}_{k\in\{1,...,K\}}$ according to Eq. (2) or Eq. (8)
    **3. Knowledge Distillation Stage**
    random shuffle the pseudo datasets $\left\{\tilde{\mathcal{D}}_k^t\right\}_{k\in\{1,...,K\}}$ according to Eq. (14)
    //update the global model through knowledge distillation.
    update the global model $\omega^t$ according to Eq. (15)
  **else**
    //update the global model through simple local models aggregation.
    update the global model $\omega^t \leftarrow \sum_{k\in S} \frac{N_k}{N}\omega_k^t$
  **end if**
**end for**

## 3.1 Federated Learning Stage

Under the Non-IID federated learning setting, we consider $K$ clients and one server. Let $\mathbb{C}$ be the set of all clients and $|\mathbb{C}| = K$. The $k$-th client has the local dataset $\mathcal{D}_k = \left\{\left(x_k^i, y_k^i\right)\right\}_{i=1}^{N_k}$, $N_k$ represents the number of data owned by the $k$-th client. $N$ represents the total number of data owned by all clients. Due to the Non-IID setting, the dataset $\mathcal{D}_k$ owned by each client $k \in \mathbb{C}$ is heterogeneously distributed. We define $\omega$ as the global model parameter of the server. In general, federated learning can be formulated as the following problem:

$$\min_{\omega} \frac{1}{K}\sum_{k=1}^{K} f_k\left(\omega\right), \ f_k\left(\omega\right) = \frac{1}{N_k}\sum_{i=1}^{N_k} \mathcal{L}\left(x_k^i, y_k^i; \omega\right), \tag{1}$$

where $\mathcal{L}$ is the loss fuction (e.g., cross-entropy) to measure training error.

At each Federated Learning Stage, FedF$^2$DG randomly selects a set of clients $S \subseteq \mathbb{C}$ and broadcasts the global model $\omega$ to them. Each client $k \in S$ initializes the local model with the global model and trains it by $min_{\omega} f_k(\omega)$. The server then collects the local models $\{\omega_k\}_{k \in S}$ and updates the global model by aggregating them with the average gradient before the incorporating round $I$, while after the incorporating round $I$, the global model is updated at Knowledge Distillation Stage.

Since the significant drift among local models in the Non-IID setting and the limited capabilities of simple aggregation employed in the Federated Learning Stage, deep knowledge from clients cannot be effectively learned, resulting in potential loss of useful knowledge. To address these issues and enable the global model to learn more useful client knowledge, we propose the following two stages: Adaptive Data Generation Stage and Knowledge Distillation Stage.

## 3.2 Adaptive Data Generation Stage

The knowledge distillation method relies on a public dataset, so some previous data-dependent knowledge distillation methods for FL [18–22] typically assume the existence of a public dataset on the server to fulfill this requirement. Nevertheless, many scenarios in FL do not align with this assumption. As a solution, we propose Adaptive Data Generation Stage, which is utilized to generate pseudo datasets containing client knowledge for the later Knowledge Distillation Stage.

And some data-free knowledge distillation methods for FL [23, 24, 33] relies on a data generator to generate pseudo dataset, which greatly relies on the server's computational resources and fails to utilize the clients' free computational resources. Therefore, in Section 3.2.1, we found a generator-free data generation method that only utilizes the local models of the clients. Since the local model is stored on both the client and the server, we can allocate the pseudo data generation tasks by observing the current computational resource usage of the client and server in real time. In addition, the quality of the generated data is also more stable because we conduct several rounds of Federated Learning Stage individually to obtain local models with excellent performance before incorporating Adaptive Data Generation Stage.

In FL, different clients possess varying amounts of information, i.e., different label distributions and numbers of local datasets. Additionally, the performance of the client's local model dynamically changes in each communication round. Hence, We propose Data Generation Principle in Section 3.2.2 to adaptively generate different numbers and label distributions of pseudo datasets for different clients respectively based on their current states in each communication round.

### 3.2.1 Data Generation Method

Inspired by DeepInversion [31], a method for synthesizing images from image distribution, we are able to optimize the noise into an image similar to dataset $\mathcal{X}$ and only require a model $\theta$ trained on dataset $\mathcal{X}$. Given a randomly initialized input ($\tilde{x} \in \mathcal{R}^{H \times W \times C}, H, W, C$ being the height, width, and number of color channels) and

an arbitrary target label $y$, the image is synthesized by optimizing:

$$\min_{\tilde{x}} \mathcal{L}\left(\tilde{x}, y; \theta\right) + \mathcal{R}_{\mathrm{DI}}\left(\tilde{x}\right), \tag{2}$$

$$\mathcal{R}_{\mathrm{DI}}\left(\tilde{x}\right) = \mathcal{R}_{prior}\left(\tilde{x}\right) + \alpha_f \mathcal{R}_{feature}\left(\tilde{x}\right), \tag{3}$$

where $\mathcal{L}\left(\cdot\right)$ is a classfication loss (e.g., cross-entropy) and $\mathcal{R}_{\mathrm{DI}}\left(\cdot\right)$ is an image regularization term composed of $\mathcal{R}_{prior}\left(\cdot\right)$ and $\mathcal{R}_{feature}\left(\cdot\right)$. $\alpha_f$ is a hyperparameter controlling the proportion of $\mathcal{R}_{feature}\left(\cdot\right)$. $\mathcal{R}_{prior}\left(\cdot\right)$ is employed to facilitate the convergence of $\tilde{x}$ towards a valid image:

$$\mathcal{R}_{prior}\left(\tilde{x}\right) = \alpha_{\mathrm{tv}} \mathcal{R}_{\mathrm{TV}}\left(\tilde{x}\right) + \alpha_{\ell_2} \mathcal{R}_{\ell_2}\left(\tilde{x}\right), \tag{4}$$

where $\mathcal{R}_{\mathrm{TV}}\left(\cdot\right)$ and $\mathcal{R}_{\ell_2}\left(\cdot\right)$ respectively penalize the total variance and $\ell_2$ norm of $\tilde{x}$, and are scaled by factors $\alpha_{\mathrm{tv}}$ and $\alpha_{\ell_2}$. $\mathcal{R}_{feature}\left(\cdot\right)$ is employed to make the feature distribution of $\tilde{x}$ at all levels similar to the natural (or original training) image $x \in \mathcal{X}$ :

$$\begin{aligned} \mathcal{R}_{feature}\left(\tilde{x}\right) = & \sum_{l} ||\mu_l\left(\tilde{x}\right) - \mathbb{E}\left(\mu_l\left(x\right)|\mathcal{X}\right)||_2 + \\ & \sum_{l} ||\sigma_l^2\left(\tilde{x}\right) - \mathbb{E}\left(\sigma_l^2\left(x\right)|\mathcal{X}\right)||_2, \end{aligned} \tag{5}$$

where $\mu_l\left(\tilde{x}\right)$ and $\sigma_l^2\left(\tilde{x}\right)$ are the batch-wise mean and variance estimates of feature maps corresponding to the $l$-th convolutional layer of model $\theta$. The $\mathbb{E}\left(\cdot\right)$ and $||\cdot||_2$ operators denote the expected value and $\ell_2$ norm calculations, respectively. $\mathbb{E}\left(\mu_l\left(x\right)|\mathcal{X}\right)$ and $\mathbb{E}\left(\sigma_l^2\left(x\right)|\mathcal{X}\right)$ can be estimated by approximately as:

$$\mathbb{E}\left(\mu_l\left(x\right)|\mathcal{X}\right) \simeq \mathrm{BN}_l(\mathrm{running\_mean}), \tag{6}$$

$$\mathbb{E}\left(\sigma_l^2\left(x\right)|\mathcal{X}\right) \simeq \mathrm{BN}_l(\mathrm{running\_variance}), \tag{7}$$

where $\mathrm{BN}_l\left(\cdot\right)$ denotes the running average statistics stored in the $l$-th BatchNorm(BN) layer of model $\theta$ during the process of training the model $\theta$ using dataset $\mathcal{X}$. Thus we can obtain $\mathbb{E}\left(\mu_l\left(x\right)|\mathcal{X}\right)$ and $\mathbb{E}\left(\sigma_l^2\left(x\right)|\mathcal{X}\right)$ from the $l$-th BN layer of model $\theta$.

At each Adaptive Data Generation Stage, FedF$^2$DG lets each client $k \in \mathbb{C}$ generate the pseudo dataset $\tilde{\mathcal{D}}_k$ by Eq. (2) using the local model $\omega_k$ that has been trained with the local dataset $\mathcal{D}_k$. Since this data generation method can generate pseudo datasets with different feature distribution based on different local models, we can regard this data generation method as adaptive. In addition, this data generation method only requires local models, thus we can choose whether to generate pseudo datasets on the clients or on the server, depending on the current availability of computational resources.

In order to generate images of diversity, [25, 35] have proposed to encourage the synthesized images to cause local model-global model disagreement. An additional regularization term $\mathcal{R}_{\mathrm{compete}}$ is added to Eq. (2) based on the Jenson-Shannon divergence

that penalizes output distribution similarities:

$$\min_{\tilde{x}} \mathcal{L}\left(\tilde{x}, y; \omega_k\right) + \mathcal{R}_{\mathrm{DI}}\left(\tilde{x}\right) + \alpha_c \mathcal{R}_{\mathrm{compete}}\left(\tilde{x}\right), \tag{8}$$

$$\mathcal{R}_{\mathrm{compete}}\left(\tilde{x}\right) = 1 - \mathrm{JS}\left(\omega_k\left(\tilde{x}\right), \omega\left(\tilde{x}\right)\right), \tag{9}$$

$$\mathrm{JS}\left(\omega_k\left(\tilde{x}\right), \omega\left(\tilde{x}\right)\right) = \frac{1}{2}\left(\mathrm{KL}\left(\omega_k\left(\tilde{x}\right), M\right) + \mathrm{KL}\left(\omega\left(\tilde{x}\right), M\right)\right),$$

where $\alpha_c$ is a hyperparameter controlling the proportion of $\mathcal{R}_{\mathrm{compete}}$. $\mathrm{KL}\left(\cdot\right)$ denotes the Kullback-Leibler divergence and $\omega_k\left(\tilde{x}\right)$ and $\omega\left(\tilde{x}\right)$ are the output distributions produced by the local and global model respectively. $M = \frac{1}{2} \cdot \left(\omega_k\left(\tilde{x}\right) + \omega\left(\tilde{x}\right)\right)$ is the average of the local and global model distributions.

During optimization, this new regularization term can help generate the hard samples [36–38] in data distribution that the global model cannot correctly classify, while the local model can. This term enhances image diversity by exploiting disagreements between local model and global model, enabling the constantly updated global model to learn local knowledge that has not yet been learned.

### 3.2.2 Data Generation Principle

To adaptively adjust the label distribution and number of pseudo dataset generated by each client in each communication round according to the current state of the client, aiming to better incorporate the knowledge of each client in each communication round into the generated pseudo datasets, we propose the following Data Generation Principle.

**Adaptively Determining the Number of Generated Dataset.** Generally speaking, the more data a client has, the more knowledge it possesses. Therefore we should let clients with more data generate more pseudo data and clients with less data generate less pseudo data. We use the number of data $N_k$ as a factor that influences the number of the pseudo dataset $\tilde{\mathcal{D}}_k$.

Additionally, we can indirectly understand the knowledge that the global model has learned so far based on its performance on the pseudo datasets generated at previous Adaptive Data Generation Stage, and then we can use it as another factor that influences the number of the pseudo dataset $\tilde{\mathcal{D}}_k$ at current stage. Specifically, after generating the pseudo datasets, we record the global model's loss on each pseudo dataset. We argue that a large loss of the global model on the pseudo dataset $\tilde{\mathcal{D}}_k^{t-1}$ at the previous stage $t-1$ indicates that the global model has not learned enough knowledge regarding client $k$. As a result, the knowledge of client $k$ is needed for the global model at the current stage $t$. Therefore, we propose the client quality $Q_k^t$ which is determined based on the $loss_k^{t-1}$, which represents the loss of the global model $\omega^{t-1}$ on the pseudo dataset $\tilde{\mathcal{D}}_k^{t-1}$:

$$Q_k^t = e^{\alpha loss_k^{t-1}}, \tag{10}$$

where $\alpha$ is a hyperparameter for normalization.

According to the above said, at $t$-th Adaptive Data Generation Stage, the number of the pseudo dataset $\tilde{\mathcal{D}}_k^t$ generated by each client $k$ is determined adaptively by the

following two factors: the number of data $N_k$ and the client quality $Q_k^t$. Given the total number of data $\tilde{\mathcal{N}}_{total}^t$ to be generated at $t$-th stage, the number of the pseudo dataset $\tilde{\mathcal{D}}_k^t$ to be generated by client $k$ is:

$$\tilde{\mathcal{N}}_k^t = \tilde{\mathcal{N}}_{total}^t * (\lambda \frac{N_k}{\sum_j N_j} + (1-\lambda) \frac{Q_k^t}{\sum_j Q_j^t}), \tag{11}$$

where $\lambda$ is a hyperparameter that controls the weights of $N_k$ and $Q_k^t$. According to Eq. (11), the number of the pseudo dataset generated by each client can be adaptively adjusted at each Adaptive Data Generation stage, so that the global model can more fully learn the knowledge of each client.

**Adaptively Sampling Label.** Under the Non-IID setting, label distributions are different among clients, i.e., $p^i(y) \neq p^j(y)$ for different clients $i$ and $j$. Since the label distribution is skewed, each local model will perform well on some classes and poorly on others. Namely, for different local models, the importance of one class is different. It is obvious that a client's knowledge should be mainly focused on its majority classes. Therefore, we cannot generate each pseudo dataset by uniformly sampling class labels. Because it will generate many minority classes data containing few of knowledge, which will lead to a decrease in the effectiveness of knowledge distillation. To address this issue, we customize the label distribution $\tilde{p}^k(y)$ for the pseudo dataset $\tilde{\mathcal{D}}_k$ based on the label distribution $p^k(y)$ of the local dataset for each client $k$, in order to generate more pseudo data with useful information,

$$\tilde{p}^k(y) = p^k(y). \tag{12}$$

According to Eq. (12), each pseudo dataset has high probability of generating the majority classes of its corresponding client, thus FedF$^2$DG can guarantee that the global model can focus on learning useful information about each client in the Non-IID setting.

## 3.3 Knowledge Distillation Stage

After Adaptive Data Generation Stage, Knowledge Distillation Stage is entered to enable the global model to fully learn the knowledge from the pseudo datasets.

At $t$-th Knowledge Distillation Stage, FedF$^2$DG utilizes the pseudo datasets $\left\{\tilde{\mathcal{D}}_k^t\right\}_{k\in\{1,\ldots,K\}}$, local models $\{\omega_k^t\}_{k\in\{1,\ldots,K\}}$ and the previous global model $\omega^{t-1}$ to perform knowledge distillation and obtain a better global model $\omega^t$. Specifically, for each pseudo dataset $\tilde{\mathcal{D}}_i^t$, we use the global model $\omega^{t-1}$ as the student model and the local model $\omega_i^t$ as the teacher model, the parameters of the global model $\omega^t$ can be learned by

$$\min_{\omega^{t-1}} \sum_{i=1}^{K} \sum_{x\in\tilde{\mathcal{D}}_i^t} \mathrm{KL}\left(\omega_i^t(x), \omega^{t-1}(x)\right), \tag{13}$$

where $\mathrm{KL}(\cdot)$ denotes the Kullback-Leibler divergence, $\omega_i^t(x)$ and $\omega^{t-1}(x)$ are the output distributions produced by the local and global model respectively, typically

obtained using a high temperature on the softmax inputs.

Note that at Knowledge Distillation Stage, if the global model sequentially utilizes the pseudo dataset on each client for knowledge distillation, it may lead to the global model overfitting to a single client's local model, resulting in the performance degradation. Table 6 shows the result of knowledge distillation of the global model sequentially utilizing the pseudo dataset on each client. To address this issue, FedF$^2$DG proposes to perform random shuffling on all pseudo datasets before conducting knowledge distillation:

$$\tilde{\mathcal{D}}^t_{shuffle} = Random\_Shuffling(\bigcup_{k=1}^{K} \tilde{\mathcal{D}}^t_k), \tag{14}$$

$$\min_{\omega^{t-1}} \sum_{x \in \tilde{\mathcal{D}}^t_{shuffle}} \mathrm{KL}\left(\omega_T\left(x\right), \omega^{t-1}\left(x\right)\right), \tag{15}$$

the teacher model $\omega_T$ corresponding to each pseudo data $x \in \tilde{\mathcal{D}}^t_{shuffle}$ is the local model of the client that generated it. FedF$^2$DG employs random shuffling to ensure that the global model can equally learn knowledge from each local model, instead of overfitting to a single local model's knowledge.

# 4 Experiments

In this section, we compare the performance of our proposed method FedF$^2$DG with other key related work.

## 4.1 Experimental Setup

The basic experimental setting is as follows.

**Baselines.** We compare FedF$^2$DG with several baseline methods FedAvg [1], FedProx [9], SCAFFOLD [12], FedNova [14] and MOON [13], with two data-dependent knowledge distillation methods FedMD [19], FedDF [18] and with a data-free knowledge distillation method FedGen [23].

**Datasets.** We conduct experiments on three benchmark datasets: CIFAR10, CIFAR100 [39] and SVHN [40], which are three difficult tasks in FL scenario and are widely adopted in FL research. Reference to existing works [10, 24], we use Dirichlet distribution $\mathbf{Dir}(\beta)$ to partition the above three datasets, thereby simulating the Non-IID data distribution among clients. A smaller $\beta$ indicates higher data heterogeneity. During the implementation, we set $\beta = 0.05$ and $\beta = 0.1$ to simulate FL scenario with large data heterogeneity. Also, we conduct experiments in FL scenario with IID data distribution. We use ResNet-18 [41] as the basic backbone for the three datasets for the following reasons: 1) The regularization term $\mathcal{R}_{feature}$ in Data Generation Method requires the model to have Batch Normalization (BN) layers. 2) ResNet-18 is a commonly used model for these three datasets.

**Hyperparameters.** The hyperparameters of the experiment are as follows:

communication round $T = 100$, incorporating round $I = 80$, the client number $K = 10, 20$ corresponding to the active fraction $F = 1, 0.5$ respectively, local training epoch $E = 5$, the local training batchsize is 64, the learning rates for classifier is 0.01

and the weight decay is 1e-5. Typically, We adopt SCAFFOLD as the FL optimizer in FedF$^2$DG.

For data generation in FedF$^2$DG, we adopt Adam optimizer with learning rate 0.1. We set $\alpha_f = 1$, $\alpha_{\text{tv}} = 0.001$, and $\alpha_{\ell_2} = 0$. After incorporating round $I = 80$ and before the 90-th round, we set $\alpha_c = 0.0$ to generate data by Eq. (2). After the 90-th round, we set $\alpha_c = 10.0$ to generate hard samples by Eq. (8). We set $\alpha = 1$ to compute the client quality, total number of data generated in $t$-th round $\tilde{\mathcal{N}}^t_{total} = 2560$, and $\lambda = 2/3$ to controls the weights of quantity and quality. For knowledge distillation in FedF$^2$DG, we adopt SGD optimizer with momentum 0.9, learning rate 0.01, and weight decay 1e-4, knowledge distillation temperature 3.

## 4.2 Performance Comparison

The performance of our method FedF$^2$DG is compared with other methods as follows:

**Test Accuracy.** Table 2, Table 3 and Table 4 report the test accuracy of all compared algorithms on CIFAR10, CIFAR100 and SVHN datasets, respectively.

On CIFAR10, FedF$^2$DG achieves the best performance in all scenarios. In data heterogeneity scenarios ($\beta = 0.1$ and $\beta$=0.05) with 10 clients, FedF$^2$DG exceeds the suboptimal method by 1.73% and 6.47% respectively. And with 20 clients, which better simulates the data distribution of real scenario, i.e. the data is distributed across more clients for federated training, FedF$^2$DG is more effective, outperforming the suboptimal method by 2.47% and 10.41% respectively.

On CIFAR100, FedF$^2$DG achieves the best performance in all scenarios. In data heterogeneity scenarios with 10 clients, FedF$^2$DG exceeds the second one by 2.36% and 4.87% respectively. And with 20 clients, outperforming the second one by 4.39% and 6.16% respectively.

On SVHN, where all methods achieved high test accuracy, FedF$^2$DG still achieves the best performance in all scenarios. In data heterogeneity scenarios with 10 clients, FedF$^2$DG exceeds the second one by 0.91% and 1.01% respectively. And with 20 clients, outperforming the second one by 0.63% and 2.41% respectively.

From the above observation, FedF$^2$DG can achieve SOTA performance even in the scenario where the data is distributed across multiple clients and is extremely heterogeneous. Besides, FedF$^2$DG outperforms the existing knowledge distillation methods FedMD, FedDF and FedGen in all scenarios, especially extremely heterogeneous scenarios where the performance of the other methods drops significantly. Finally, the excellent performance of FedF$^2$DG validates the effectiveness of enhancing the model aggregation step by data-free knowledge distillation.

**Convergence of FedF$^2$DG.** Figure 2 displays the convergence curve of FedF$^2$DG in three datasets with different data heterogeneity $\beta$, it can be seen that FedF$^2$DG reaches convergence in all settings. Following the incorporating round $I = 80$, we added the second and third stages: Adaptive Data Generation Stage and Knowledge Distillation Stage. This led to an increase in test accuracy at 80 round, followed by a gradual convergence.

**Classification results for FedF$^2$DG.** Figure 3 displays the classification results of FedF$^2$DG in three datasets with different data distribution settings. As can be seen from the clear diagonal lines of the confusion matrix, FedF$^2$DG achieves relatively

**Table 2**: Test accuracy (%) of different methods on CIFAR10 with 10 and 20 clients. The content in ( ) indicates the percentage improvement in the accuracy of FedF$^2$DG compared to the suboptimal method.

| | 10 clients | | | 20 clients | | |
|---|---|---|---|---|---|---|
| | IID | $\beta = 0.1$ | $\beta = 0.05$ | IID | $\beta = 0.1$ | $\beta = 0.05$ |
| FedAvg | 93.51 | 86.25 | 75.82 | 92.38 | 80.73 | 63.71 |
| FedProx | 93.74 | 85.67 | 77.62 | 92.41 | 82.52 | 71.38 |
| FedNova | 93.63 | 87.52 | 68.68 | 92.34 | 81.12 | 63.95 |
| MOON | 92.65 | 85.58 | 77.19 | 91.56 | 80.25 | 63.69 |
| SCAFFOLD | 93.86 | 87.01 | 78.05 | 93.13 | 85.23 | 72.97 |
| FedMD | 84.02 | 81.63 | 80.51 | 81.24 | 73.06 | 72.80 |
| FedDF | 92.03 | 71.02 | 65.64 | 88.63 | 67.69 | 36.90 |
| FedGen | 88.35 | 81.51 | 78.90 | 87.78 | 72.80 | 71.65 |
| FedF$^2$DG | **93.90**(+0.04%) | **89.04**(+1.73%) | **85.72**(+6.47%) | **93.21**(+0.08%) | **87.34**(+2.47%) | **80.57**(+10.41%) |

**Table 3**: Test accuracy (%) of different methods on CIFAR100 with 10 and 20 clients. The content in ( ) indicates the percentage improvement in the accuracy of FedF$^2$DG compared to the suboptimal method.

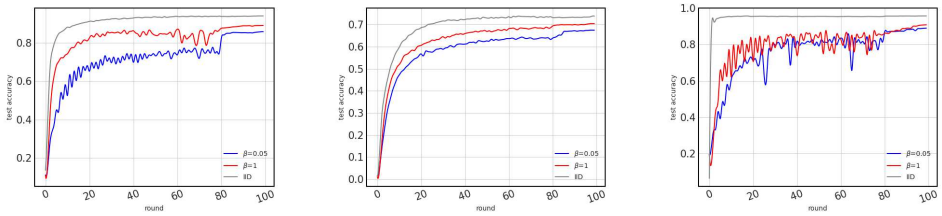
| | 10 clients | | | 20 clients | | |
|---|---|---|---|---|---|---|
| | IID | $\beta = 0.1$ | $\beta = 0.05$ | IID | $\beta = 0.1$ | $\beta = 0.05$ |
| FedAvg | 71.34 | 67.02 | 63.70 | 69.11 | 62.21 | 59.60 |
| FedProx | 71.67 | 66.74 | 63.49 | 69.26 | 62.50 | 59.87 |
| FedNova | 71.01 | 67.19 | 64.30 | 68.80 | 62.25 | 59.15 |
| MOON | 69.04 | 66.75 | 63.18 | 65.86 | 62.12 | 59.48 |
| SCAFFOLD | 73.81 | 68.83 | 64.35 | 72.86 | 65.31 | 61.76 |
| FedMD | 45.86 | 19.84 | 17.49 | 20.18 | 12.17 | 13.97 |
| FedDF | 65.69 | 51.25 | 40.96 | 57.03 | 37.98 | 29.16 |
| FedGen | 33.08 | 28.16 | 26.27 | 30.05 | 24.40 | 19.09 |
| FedF$^2$DG | **73.84**(+0.04%) | **70.46**(+2.36%) | **67.49**(+4.87%) | **73.07**(+0.28%) | **68.18**(+4.39%) | **65.57**(+6.16%) |

**Table 4**: Test accuracy (%) of different methods on SVHN with 10 and 20 clients. The content in ( ) indicates the percentage improvement in the accuracy of FedF$^2$DG compared to the suboptimal method.

| | 10 clients | | | 20 clients | | |
|---|---|---|---|---|---|---|
| | IID | $\beta = 0.1$ | $\beta = 0.05$ | IID | $\beta = 0.1$ | $\beta = 0.05$ |
| FedAvg | 95.01 | 86.45 | 83.37 | 94.82 | 91.02 | 83.79 |
| FedProx | 95.01 | 87.03 | 84.84 | 94.80 | 91.61 | 88.22 |
| FedNova | 95.00 | 89.95 | 85.33 | 95.03 | 91.95 | 88.02 |
| MOON | 94.04 | 87.51 | 81.32 | 93.83 | 90.86 | 81.42 |
| SCAFFOLD | 95.64 | 85.54 | 82.35 | 95.42 | 91.93 | 84.17 |
| FedMD | 92.02 | 89.37 | 88.07 | 90.59 | 89.40 | 89.00 |
| FedDF | 95.22 | 76.57 | 34.85 | 94.64 | 88.72 | 65.62 |
| FedGen | 90.37 | 87.75 | 87.32 | 89.66 | 88.46 | 87.91 |
| FedF$^2$DG | **95.65**(+0.01%) | **90.77**(+0.91%) | **88.96**(+1.01%) | **95.50**(+0.08%) | **92.53**(+0.63%) | **91.15** (+2.41%) |

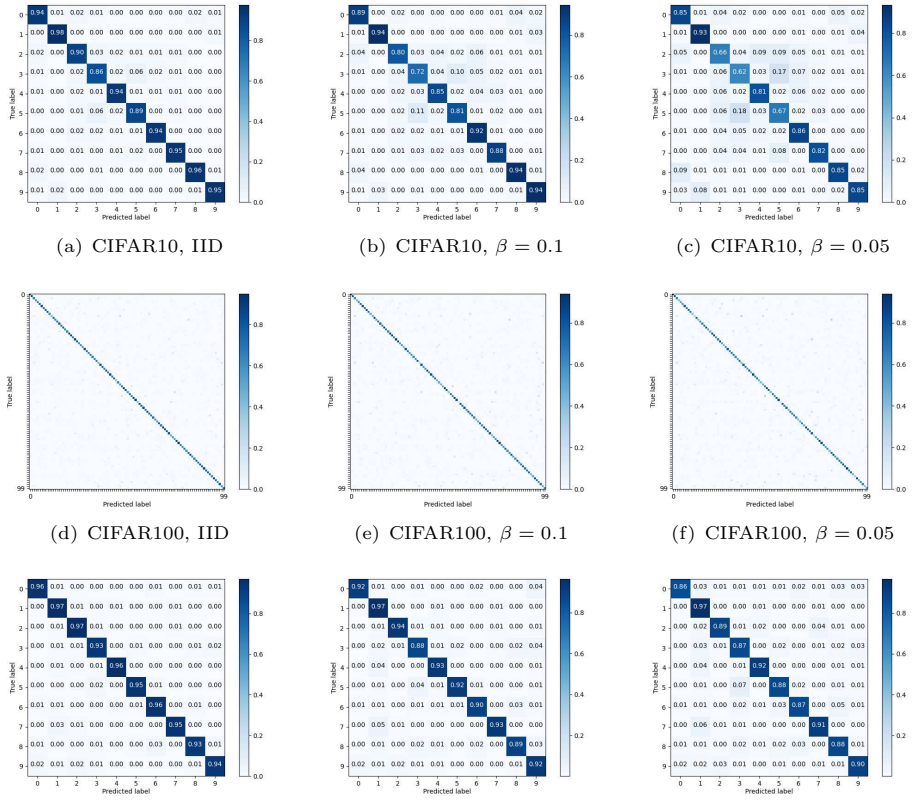
good classification results for each class in all scenarios.

**Data heterogeneity.** Figure 4 displays the test accuracy of different FL methods on different $\beta$ values. FedF$^2$DG achieves SOTA performance in all settings, which proofs that FedF$^2$DG can help the global model to improve performance in all data heterogeneity scenarios. In addition, FedF$^2$DG demonstrates significantly higher accuracy compared to other methods in the case of extreme data heterogeneity ($\beta = 0.05$).




(a) CIFAR10 (b) CIFAR100 (c) SVHN

**Fig. 2**: convergence curve of FedF$^2$DG in three datasets with different data heterogeneity $\beta$, in 10 clients scenario.




(a) CIFAR10, IID (b) CIFAR10, $\beta = 0.1$ (c) CIFAR10, $\beta = 0.05$

(d) CIFAR100, IID (e) CIFAR100, $\beta = 0.1$ (f) CIFAR100, $\beta = 0.05$

(g) SVHN, IID (h) SVHN, $\beta = 0.1$ (i) SVHN, $\beta = 0.05$

**Fig. 3**: Confusion matrix of FedF$^2$DG in three datasets with different data distributions, in 20 clients scenario.

And as the degree of data heterogeneity decreases ($\beta$ increases), the test accuracy of each method rises. However, FedF$^2$DG still significantly outperforms other methods and outperforms SCAFFOLD.

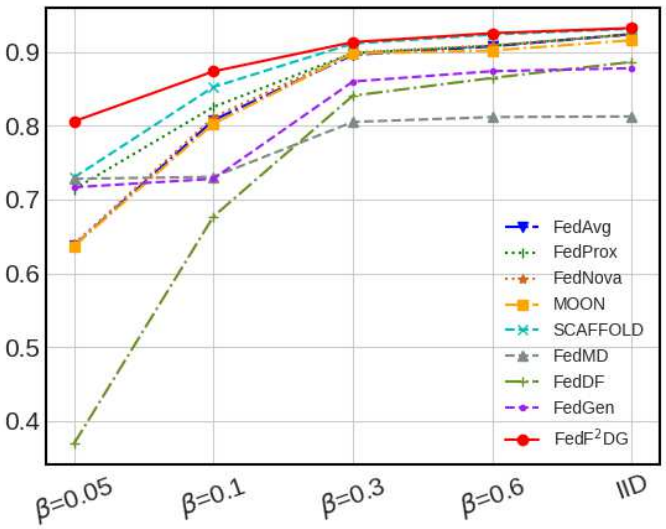



**Fig. 4**: Test accuracy w.r.t. data heterogeneity $\beta$. Experiments were conducted on CIFAR10 in 20 clients scenario.

**The effect of FedF$^2$DG combined with existing FL optimizers.** Table 5 shows the effect of FedF$^2$DG combined with FedAvg, FedProx, FedNova, MOON and SCAFFOLD. In Table 5, FedProx+FedF$^2$DG and SCAFFOLD+FedF$^2$DG respectively show the best test accuracy among all optimizers with $\beta = 0.05$ and $\beta = 0.1$. Comparing Table 5 with Table 2, we observe that the performance of any FL optimizer can be greatly boosted by combining it with FedF$^2$DG. This validates the effectiveness and combinability of FedF$^2$DG. Furthermore, simply using FedAvg as local optimizer in combination with FedF$^2$DG (FedAvg+FedF$^2$DG) already outperforms the other methods in Table 2.

## 4.3 Ablation Study

**Necessity of each component in FedF$^2$DG.** Table 6 shows the performance of FedF$^2$DG with the loss of some modules. The specific implementation of each module is as follows:

- -$\mathcal{R}_{\text{compete}}$: after the incorporating round $I$, i.e. after incorporating the two stages of Adaptive Data Generation and Knowledge Distillation, only Eq.(2) is used to generate data, while the regularization term $\mathcal{R}_{\text{compete}}$ (Eq.(8)) is not used to generate hard samples.

**Table 5**: The effect of FedF$^2$DG combined with different FL optimizers. Test accuracy (%) on CIFAR10, $\beta = 0.1$ and 0.05 in 20 clients scenario. The content in ( ) indicates the percentage improvement in accuracy when combined with FedF$^2$DG compared to the original FL optimizer.

| | Accuracy(%) | |
|---|---|---|
| | $\beta = 0.1$ | $\beta = 0.05$ |
| FedAvg+FedF$^2$DG | 86.34 (+6.94%) | 78.22 (+22.77%) |
| FedProx+FedF$^2$DG | 86.63 (+4.98%) | **80.57** (+12.87%) |
| FedNova+FedF$^2$DG | 86.25 (+6.32%) | 78.27 (+22.39%) |
| MOON+FedF$^2$DG | 86.01 (+7.17%) | 78.12 (+22.65%) |
| SCAFFOLD+FedF$^2$DG | **87.34** (+2.47%) | 79.69 (+9.20%) |

**Table 6**: Impact of the each component in FedF$^2$DG. The experiments are conducted on CIFAR10, $\beta = 0.05$ in 20 clients scenario. The content in ( ) indicates the percentage decrease in accuracy compared to baseline method.

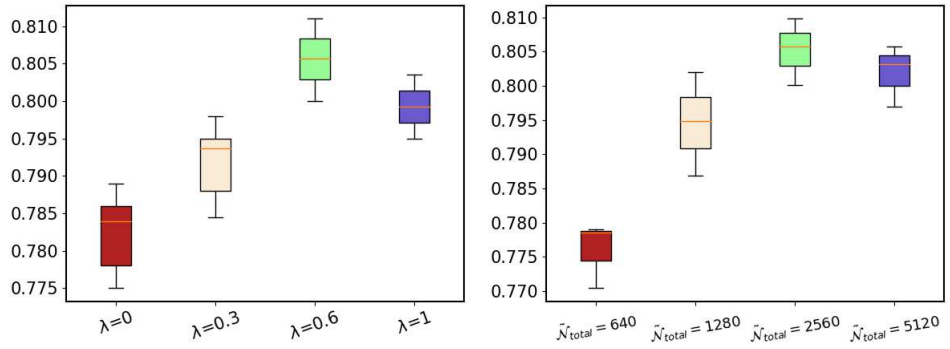
| | Method | Accuracy(%) |
|---|---|---|
| baseline | FedF$^2$DG | **80.57** |
| module | - $\mathcal{R}_{\text{compete}}$ | 79.19 (-1.71%) |
| | - ADNGD | 79.06 (-1.87%) |
| | - ASL | 76.86 (-4.60%) |
| | - DGP | 76.01 (-5.65%) |
| | - Random shuffling | 73.28 (-9.04%) |

- -ADNGD: each client generates a fixed number of 128 data per round (total of 2560 data generated per round for all clients.) without using Adaptive Determining the Number of Generated Dataset in the data generation principle.
- -ASL: each client obtains the label distribution of pseudo dataset through random uniform sampling without using Adaptively Sampling Label in the data generation principle.
- -DGP: including -ADNGD and -ASL.
- -Random shuffling: the global model sequentially utilizes the pseudo dataset on each client for knowledge distillation, rather than all pseudo datasets being performed random shuffling and then perform knowledge distillation.

We can observe that removing any module will result in a decrease in the performance of FedF$^2$DG. Besides, the joint absence of modules can lead to further performance degradation. The most significant performance degradation is seen in -ASL and -Random shuffling. This indicates that sampling the label distribution of pseudo dataset based on the label distribution of the client's local data can better extract knowledge from the client, and that global model overfitting can be prevented by random shuffling. Table 6 proves that each module we designed is necessary and reasonable for FedF$^2$DG.

**Sensitivity of FedF$^2$DG to hyperparameter $\lambda$.** To measure the influence of hyperparameter $\lambda$ selection, we select $\lambda$ from [0, 0.3, 0.6, 1]. Figure 5(a) shows the test accuracy in term of the box plot, where FedF$^2$DG achieves the highest test accuracy in $\lambda = 0.6$. Meanwhile $\lambda = 0.3$ and $\lambda = 1$ achieve similar test accuracy as $\lambda = 0.6$. This indicates that FedF$^2$DG is not sensitive to the selection of hyperparameter $\lambda$ in a certain range. However, when $\lambda = 0$, the test accuracy decreases more, indicating that we cannot use client quality alone as a metric to determine the number of generated data. Nevertheless, the worst test accuracy in Figure 5(a) is still better than the previous best work in Table 2.

**Influence of hyperparameter $\tilde{\mathcal{N}}_{total}$ in FedF$^2$DG.** To measure the influence of the hyperparameter $\tilde{\mathcal{N}}_{total}$ selection, we select $\tilde{\mathcal{N}}_{total}$ from [640, 1280, 2560, 5120]. In Figure 5(b), we can see that when $\tilde{\mathcal{N}}_{total}$ is small, the test accuracy rises rapidly. And when $\tilde{\mathcal{N}}_{total}$ is more than 2560, the test accuracy rises slowly and nearly smoothly. This shows that sufficient client knowledge can be extracted in the pseudo datasets when the total number $\tilde{\mathcal{N}}_{total}$ of data generated per round is equal to 2560.




(a) Different $\lambda$

(b) Different $\tilde{\mathcal{N}}_{total}$

**Fig. 5**: Influence of different hyperparameters in FedF$^2$DG. Experiments were conducted on CIFAR10, $\beta = 0.05$ in 20 clients scenario.

**Influence of incorporating round $I$ in FedF$^2$DG.** Figure 6 shows the final performance of FedF$^2$DG under different incorporating round $I$. When we added our Adaptive Data Generation Stage and Knowledge Distillation Stage earlier ($I = 0$), it is hard to fully utilize these two stages because of the poor performance of the local model, which generates low quality pseudo data. Therefore, we should add these two stages when the performance of the local model is stable ($I = 80$).

# 5 Discussion

**Privacy issue.** It is well known that user privacy is a paramount issue for FL. Since FedF$^2$DG may generate client pseudo datasets on the server, it violates the privacy regulations in FL. However, if we generate pseudo datasets locally on each client, we will

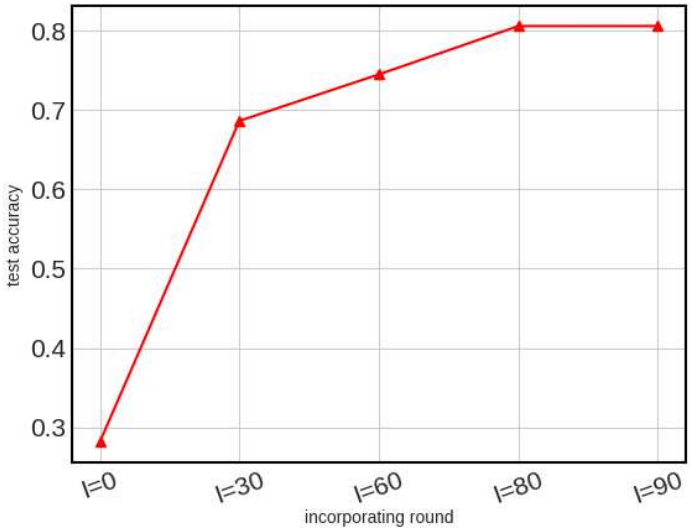



**Fig. 6**: Test accuracy of FedF$^2$DG under different incorporating round $I$. Experiments were conducted on CIFAR10, $\beta = 0.05$ in 20 clients scenario.

not compromise user privacy, although this requires sufficient computational resources on the client. In our experiments, for convenience, we upload all pseudo datasets to the server to perform random shuffling. Nevertheless, we can protect privacy by making the server communicate with each client randomly in turn to conduct knowledge distillation, although this requires additional communication costs.

**Pseudo dataset quality.** Generally speaking, the quality of the pseudo dataset generated by our data generation method is stable but depends on the local model quality. In empirical experiments, we found that the local model quality depends on the label distribution and quantity of client local data, i.e., the quality of client local data. Therefore future work can investigate how to evaluate the clien data quality and use it to guide our Adaptive Data Generation Stage.

**Computational resources.** Computational resources are a major limitation of this work. Since FedF$^2$DG carries out additional data generation stage and knowledge distillation stage, this will make the whole training time longer than other methods and requires the server to have sufficient computational resources. However, the servers in the current FL application scenario [4, 42–44] are usually large organizations that own sufficient computational resources, so FedF$^2$DG is applicable.

# 6 Conclusion

In this paper, we propose a novel data-Free knowledge distillation approach FedF$^2$DG via generator-Free Data Generation to boost the performance of federated learning by transfering the knowledge in local models to the global model. We propose a Data

Generation Method that leverages only local models by optimizing noise into real images using a regularization term, and can generate hard samples by adding an additional regularization term that exploit disagreements between local model and global model. Meanwhile, FedF$^2$DG enables flexible utilization of computational resources by generating pseudo dataset locally or on the server. To address the label distribution shift in data heterogeneity scenario, we propose a Data Generation Principle that adaptively controls the label distribution and number of pseudo dataset according to client's information, which allows for the incorporation of a greater amount of client knowledge into the pseudo dataset. Extensive empirical experiments on three benchmarks validate the effectiveness of the proposed FedF$^2$DG.

# 7 Declarations

## 7.1 Ethical Approval.

Not applicable.

## 7.2 Competing interests.

The authors declare that they have no conflicts of interest.

## 7.3 Authors' contributions.

Siran Zhao contributed to the Conception of the study, Writing original draft, Methodology, Software, Validation, Experiment, revision of the draft and Investigation. Tianchi Liao and Lele Fu contributed to the revision of the draft; Chuan Chen, Jing Bian, and Zibin Zheng contributed to the supervision and Funding acquisition;

## 7.4 Funding.

The research is supported by the Key-Area Research and Development Program of Guangdong Province (No. 2020B1111370001), the National Natural Science Foundation of China (62176269), the Guangzhou Science and Technology Program (2023A04J0314), and the Sun Yat-Sen University Young Faculty Development Program (No. 23ptpy109).

## 7.5 Availability of data and materials.

All datasets used in this study are publicly available.

# References

[1] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282 (2017). PMLR

[2] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016

ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318 (2016)

[3] Lim, W.Y.B., Garg, S., Xiong, Z., Niyato, D., Leung, C., Miao, C., Guizani, M.: Dynamic contract design for federated learning in smart healthcare applications. IEEE Internet of Things Journal **8**(23), 16853–16862 (2020)

[4] Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F.: Federated learning for healthcare informatics. Journal of Healthcare Informatics Research **5**, 1–19 (2021)

[5] Liu, Q., Chen, C., Qin, J., Dou, Q., Heng, P.-A.: Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1013–1023 (2021)

[6] Tan, B., Liu, B., Zheng, V., Yang, Q.: A federated recommender system for online services. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 579–581 (2020)

[7] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., Ramage, D.: Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2018)

[8] Hsu, T.-M.H., Qi, H., Brown, M.: Measuring the effects of non-identical data distribution for federated visual classification. arXiv preprint arXiv:1909.06335 (2019)

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

[10] Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N., Saligrama, V.: Federated learning based on dynamic regularization. arXiv preprint arXiv:2111.04263 (2021)

[11] Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189 (2019)

[12] Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International Conference on Machine Learning, pp. 5132–5143 (2020). PMLR

[13] Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10713–10722 (2021)

[14] Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. Advances in neural information processing systems **33**, 7611–7623 (2020)

[15] Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N., Saligrama, V.: Federated learning based on dynamic regularization. arXiv preprint arXiv:2111.04263 (2021)

[16] Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-iid data silos: An experimental study. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 965–978 (2022). IEEE

[17] Singh, S.P., Jaggi, M.: Model fusion via optimal transport. Advances in Neural Information Processing Systems **33**, 22045–22055 (2020)

[18] Lin, T., Kong, L., Stich, S.U., Jaggi, M.: Ensemble distillation for robust model fusion in federated learning. Advances in Neural Information Processing Systems **33**, 2351–2363 (2020)

[19] Li, D., Wang, J.: Fedmd: Heterogenous federated learning via model distillation. arXiv preprint arXiv:1910.03581 (2019)

[20] Sattler, F., Korjakow, T., Rischke, R., Samek, W.: Fedaux: Leveraging unlabeled auxiliary data in federated learning. IEEE Transactions on Neural Networks and Learning Systems (2021)

[21] Hu, L., Yan, H., Li, L., Pan, Z., Liu, X., Zhang, Z.: Mhat: An efficient model-heterogenous aggregation training scheme for federated learning. Information Sciences **560**, 493–503 (2021)

[22] Gong, X., Sharma, A., Karanam, S., Wu, Z., Chen, T., Doermann, D., Innanje, A.: Ensemble attention distillation for privacy-preserving federated learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15076–15086 (2021)

[23] Zhu, Z., Hong, J., Zhou, J.: Data-free knowledge distillation for heterogeneous federated learning. In: International Conference on Machine Learning, pp. 12878–12889 (2021). PMLR

[24] Zhang, L., Shen, L., Ding, L., Tao, D., Duan, L.-Y.: Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10174–10183 (2022)

[25] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)

[26] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: An overview. IEEE signal processing magazine **35**(1), 53–65 (2018)

[27] Chen, H., Wang, Y., Xu, C., Yang, Z., Liu, C., Shi, B., Xu, C., Xu, C., Tian, Q.: Data-free learning of student networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3514–3522 (2019)

[28] Fang, G., Song, J., Shen, C., Wang, X., Chen, D., Song, M.: Data-free adversarial distillation. arXiv preprint arXiv:1912.11006 (2019)

[29] Lopes, R.G., Fenu, S., Starner, T.: Data-free knowledge distillation for deep neural networks. arXiv preprint arXiv:1710.07535 (2017)

[30] Nayak, G.K., Mopuri, K.R., Shaj, V., Radhakrishnan, V.B., Chakraborty, A.: Zero-shot knowledge distillation in deep networks. In: International Conference on Machine Learning, pp. 4743–4751 (2019). PMLR

[31] Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8715–8724 (2020)

[32] Fang, G., Mo, K., Wang, X., Song, J., Bei, S., Zhang, H., Song, M.: Up to 100x faster data-free knowledge distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 6597–6604 (2022)

[33] Qi, P., Zhou, X., Ding, Y., Zhang, Z., Zheng, S., Li, Z.: Fedbkd: Heterogenous federated learning via bidirectional knowledge distillation for modulation classification in iot-edge system. IEEE Journal of Selected Topics in Signal Processing (2022)

[34] Yao, D., Pan, W., Dai, Y., Wan, Y., Ding, X., Jin, H., Xu, Z., Sun, L.: Local-global knowledge distillation in heterogeneous federated learning with non-iid data. arXiv preprint arXiv:2107.00051 (2021)

[35] Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)

[36] Cui, Y., Zhou, F., Lin, Y., Belongie, S.: Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1153–1162 (2016)

[37] Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 118–126

(2015)

[38] Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2794–2802 (2015)

[39] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

[40] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)

[41] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

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

[43] Jiang, J.C., Kantarci, B., Oktug, S., Soyata, T.: Federated learning in smart city sensing: Challenges and opportunities. Sensors **20**(21), 6230 (2020)

[44] Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST) **10**(2), 1–19 (2019)