Exploring One-shot Semi-supervised Federated Learning with A Pre-trained Diffusion Model

Mingzhao Yang†, Shangchao Su†, Bin Li*, Xiangyang Xue
School of Computer Science, Fudan University
{mzyang20, scsu20, libin, xyxue}@fudan.edu.cn

ABSTRACT

Federated learning is a privacy-preserving collaborative learning approach. Recently, some studies have proposed the semi-supervised federated learning setting to handle the commonly seen real-world scenarios with labeled data on the server and unlabeled data on the clients. However, existing methods still face challenges such as high communication costs, training pressure on the client devices, and distribution differences among the server and the clients. In this paper, we introduce the powerful pre-trained diffusion models into federated learning and propose **FedDISC**, a **Fed**erated **D**iffusion-**I**nspired **S**emi-supervised **C**o-training method, to address these challenges. Specifically, we first extract prototypes from the labeled data on the server and send them to the clients. The clients then use these prototypes to predict pseudo-labels of the local data, and compute the cluster centroids and domain-specific features to represent their personalized distributions. After adding noise, the clients send these features and their corresponding pseudo-labels back to the server, which uses a pre-trained diffusion model to conditionally generate pseudo-samples complying with the client distributions and train an aggregated model on them. Our method does not require local training and only involves forward inference on the clients. Our extensive experiments on DomainNet, Openimage, and NICO++ demonstrate that the proposed FedDISC method effectively addresses the one-shot semi-supervised problem on Non-IID clients and outperforms the compared SOTA methods. We also demonstrate through visualization that it is of neglectable possibility for FedDISC to leak privacy-sensitive information of the clients.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Machine learning**.

KEYWORDS

Federated learning; One shot; Diffusion model

1 INTRODUCTION

Federated Learning (FL) [27] is a new paradigm of machine learning that allows multiple clients to perform collaborative training without sharing private data. In the typical federated learning setting, each client trains the local model on its labeled dataset and sends the trained parameters to the server in each communication round. The server then aggregates all the parameters to obtain the aggregated model, which is then sent back to the clients. Through many communication rounds, the aggregated model gradually converges.

Realistic federated learning scenarios, such as autonomous driving and mobile album classification, often involve individual users who are unwilling or unable to provide reliable annotations. This often results in the client data being unlabeled in practice. Semi-supervised federated learning [5, 15, 49] has been proposed to address this issue. In this setting, there are multiple clients with unlabeled data and a server with labeled data. Usually, data from different clients are not independent and identically distributed. The goal of semi-supervised federated learning is to obtain an aggregated model that adapts to the data distribution of all clients through collaborative learning between the server and the clients.

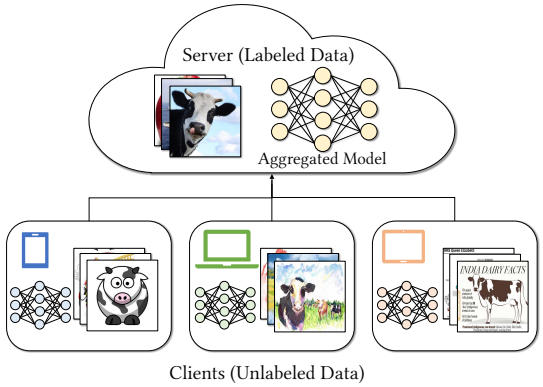

Figure 1: One-shot semi-supervised federated learning.

However, existing semi-supervised federated learning faces several challenges. First, there is a problem with the amount of communication, as traditional methods require a large amount of communication between the clients and the server, which significantly increases the burden on the clients. Second, training model is not feasible for some client devices due to hardware and computation limitations. Third, when there are distribution differences among the server and the clients, the performance of the aggregated model significantly decreases.

To address the challenge of communication, one-shot federated learning [11, 44] has been proposed, which involves the server collecting models trained by the clients and optimizing random noise based on these models to generate pseudo-samples. Unfortunately, these methods are only applicable to the scenarios where clients have sufficient data and require client training. Furthermore, when the client models cannot provide the mean and variance in batch normalization layers, the quality of the generated pseudo-samples can be poor.

Recently, the development of foundation models [31, 33] has provided us with new opportunities. On the one hand, in terms of generative models, the performance of pre-trained diffusion models on conditional image generation has become increasingly

* Corresponding author.
† Equal contribution.

impressive, which makes it possible for us to generate high-quality pseudo-samples on the server. On the other hand, for feature extraction, pre-trained CLIP, which has zero-shot classification ability, can be employed to extract features of the unlabeled client data without any training on clients.

Motivated by the above two opportunities, in this paper, we introduce **FedDISC**, a **Fed**erated **D**iffusion-**I**nspired **S**emi-supervised **C**o-training method, to leverage powerful foundation models in one-shot semi-supervised federated learning. In brief, our work consists of four steps: 1) We use the image encoder of the pre-trained CLIP to extract the prototype of each category on the server and send the encoder and the prototypes to the clients. 2) On each client, we use the received encoder to extract the features of the unlabeled client data and predict their pseudo-labels by computing the similarities between the features and the received prototypes. 3) For communication and privacy considerations, we obtain some cluster centroids and a domain-specific feature by clustering and averaging the client features for each category, and send these features to the server after adding noise. The cluster centroids represent the information of some representative images of the clients, while the domain-specific features represent the information of the overall distributions of the clients. 4) The server randomly combines the received cluster centroids and domain-specific features, and generates pseudo-samples conditioned on these features and the text prompts of the predicted pseudo-labels by a pre-trained diffusion model. With the powerful diffusion model, the generated samples exhibit a noticeable improvement in reality and diversity, which can be used in fine-tuning the downstream classification models.

Our extensive experiments on DomainNet [29], OpenImage [20], and NICO++ [48] show that the proposed FedDISC can obtain a high-performance aggregated model that adapts to various client distributions within one round communication. A large number of visualization experiments also demonstrate that we can obtain diverse and realistic images without leaking the privacy-sensitive information of the clients.

Our contributions are summarized as follows:

- We demonstrate the excellent performance of pre-trained diffusion models when applied to federated learning, which has not been explored before.
- We propose the FedDISC method, which leverages the powerful generative ability of pre-trained diffusion model to achieve a one-shot semi-supervised federated learning framework, without any client training and the large communication from iterations.
- We conduct extensive experiments on multiple real-world datasets to validate the effectiveness of FedDISC. The experimental results indicate that FedDISC effectively handles distribution differences among the server and the clients, and obtains competitive performance compared with the compared methods.

2 RELATED WORKS

2.1 Foundation Models

Recently, Fundamental Models [19, 31, 33] have achieved unprecedented success in the fields such as computer vision and natural language processing. The CLIP [31] has successfully bridged the gap between text and vision, contributing powerful pre-trained models to various downstream tasks. The diffusion models [12, 37] have provided a new generative paradigm for image generation, with Stable diffusion [33] achieving remarkable performance.

A major advantage of diffusion models is the ability to use various conditions in conditional image generation, such as a trained classifier [4], text [18, 28, 36] and images [30, 35, 41, 43, 45]. There are also some works [7, 13, 25] studying compositional visual generation with conditional diffusion models. Analyzing from the perspective of energy-based models, these works can simultaneously guide the generating process with multiple conditions by connecting different conditions through some logical operators such as AND and NOT. Generally in practice, diffusion models use Unet [34] as their main model structure.

In addition, the sampling efficiency is a challenge for diffusion models since they require iterative denoising. Some works [3, 17, 24, 26, 38, 47] focus on improving the sampling efficiency by proposing differential equations and numerical methods for diffusion models. Some works [9, 33] apply diffusion models in the latent space of a powerful autoencoder to alleviate the complexity of sampling.

From their performance in image generation, large-scale pre-trained diffusion models have a huge amount of knowledge and can conditionally generate realistic images within an acceptable cost of time and computation. This is the main reason why we apply pre-trained diffusion models in federated learning.

2.2 Federated Learning

Supervised Federated Learning. FedAvg [27] proposes the Federated Averaging method, which achieves machine learning modeling under privacy protection. However, some studies [21, 22] have indicated that in non-iid scenarios, the performance of FedAvg's aggregated model will significantly decrease. To address this challenge, numerous works have attempted to establish stronger global models capable of handling data distributions from different clients [16, 21, 32, 42], or personalized federated learning that allows clients to obtain personalized model parameters during the federated learning process [2, 6, 8, 14, 46]. In addition, to further reduce communication costs, some works [11, 39, 44] propose one-shot federated learning, which only performs one round of communication. Zhang [44] proposes to use the client models to train a generator on the server and distill the models using the generated pseudo samples. Heinbaugh [11] proposes to train conditional VAEs on the client side and generate pseudo samples using these conditional VAEs on the server. Existing one-shot federated learning requires clients to train good local models, which requires labeled data.

Semi-supervised Federated Learning. In realistic federated learning scenarios, clients often lack labeled data, but a large amount of unlabeled data is available. In response to this issue, FedMatch [15] proposes Semi-Supervised Federated Learning, which presents two different settings: the server has labeled data while the clients have unlabeled data, and the clients contain both labeled and unlabeled data. Zhang [49] focuses on the first setting, points out the importance of the gradient diversity problem, and proposes several effective strategies to suppress the diversity. Diao [5] supposes that the server has a small amount of labeled data while the clients

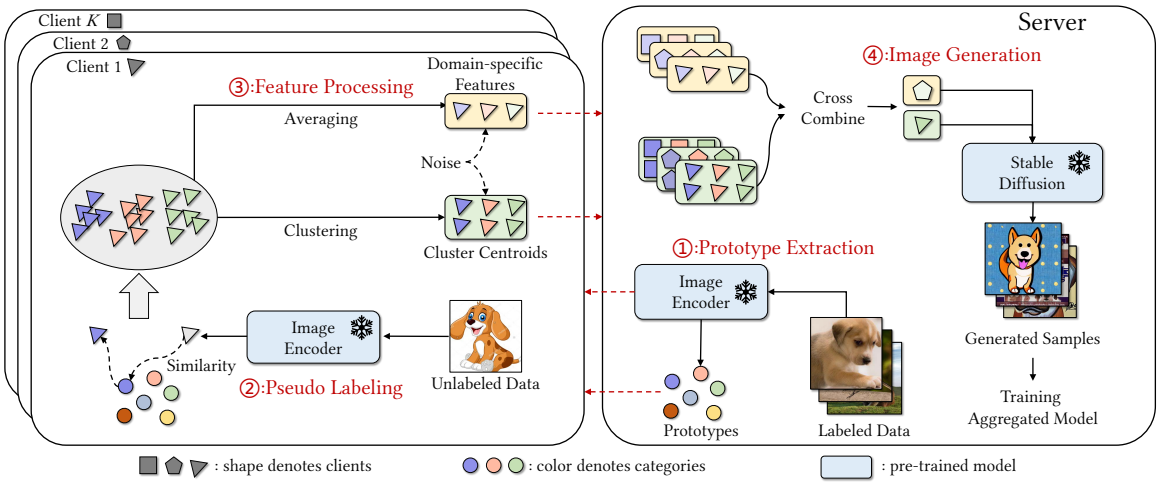

Figure 2: The framework of FedDISC. The overall method consists of four steps: Prototype Extraction, Pseudo Labeling, Feature Processing, and Image Generation.

have unlabeled data, and explores the performance of existing semi-supervised learning methods under this setting.

In the paper, we focus on a more popular setting, where the clients have only unlabeled data and the server has only labeled data. Furthermore, our additional challenge, compared to the existing semi-supervised federated learning setting, is to limit the communication rounds to only one without any training on the clients, thereby effectively controlling communication and computation costs.

2.3 FL with Foundation Models

Leveraging the powerful pre-training ability of fundamental models, some works [10, 40] in the federated learning field have explored the application of the CLIP model in federated image classification. By fine-tuning the CLIP model, these works have achieved stronger classification performance than traditional fully-trained models. However, to the best of our knowledge, no work has yet utilized pre-trained diffusion models, such as Stable Diffusion, in federated learning. In this paper, we make a novel attempt and reveal the potential of the diffusion models in federated learning. Furthermore, current foundation models cannot be trained on edge devices, making it challenging for many existing federated learning algorithms to be implemented. Fortunately, a mature technology [1] exists that allows foundation models to perform inference on low-power devices such as iPhones. Therefore, the method proposed in this paper, which does not require client-side training, has great potential for practical applications.

3 METHOD

In this section, we introduce the proposed FedDISC method, which applies powerful pre-trained diffusion models to the federated scenario, achieving a one-shot semi-supervised federated learning framework without client training. In section 3.1, we provide some notations and background knowledge on diffusion models. In section 3.2, we describe the proposed method in detail through four steps taken by the clients and the server.

[1]https://github.com/mlc-ai/mlc-llm

3.1 Preliminaries

Diffusion Models. The diffusion models study the transformation from the Gaussian distribution to the realistic distribution by iteratively executing the denoising process. In this paper, since only the pre-trained diffusion models is used and no training is conducted, the sampling process of the diffusion models is mainly introduced here. During sampling, the diffusion model ϵ_θ samples $\mathbf{x}_T$ from the Gaussian distribution, where T is the maximum timestep predetermined during training. The diffusion model takes $\mathbf{x}_T$ as the initial noise of the denoising process and uses the input text prompt p and the input image q as conditions. After T timesteps of denoising, $\mathbf{x}_T$ is restored to a real image $\mathbf{x}_0$ with specified semantics. The sampling process is as follows:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t, t|p, q)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(\mathbf{x}_t, t|p, q) + \sigma_t\varepsilon_t, t = T, .., 1$$

, where α_t, α_{t-1} and σ_t are pre-defined parameters, ε_t is the Gaussian noise randomly sampled at each timestep. It should be noted that currently many methods can freely control the number of iterations of the denoising process to accelerate sampling, but the overall process is quite similar, so these methods won't be elaborated here.

Notations and Objectives. In this paper, we consider a semi-supervised federated learning setting, where we have K clients with unlabeled datasets $\mathcal{D}_k = \{\mathbf{x}_i^k\}_{i=1}^{N_k}, k = 1, \ldots, K$, where N_k is the number of images on the k-th client, and a server with a labeled dataset $\mathcal{D}_s = \{\mathbf{x}_i^s, y_i\}_{i=1}^{N_s}, y_i \in \{0, \ldots, M\}$, where N_s is the number of images on the server and M is the number of categories. The text prompts of these categories are $C_j, j \in \{0, \ldots, M\}$. The objective

Algorithm 1 *FedDISC*: a Federated Diffusion-Inspired Semi-supervised Co-training method

Input:$\mathcal{D}_k, k = 1, \ldots, K$, $\mathcal{D}_s$, the pre-trained encoder E_θ, the pre-trained diffusion model ϵ_θ, and classifier F_θ.

1: Server sends the prototypes to clients according to Eq.2.
2: **for** local device $i = 1$ to K in parallel **do**
3: Calculate pseudo labels using Eq.3.
4: Perform feature processing, including clustering, averaging, and adding noise using Eq.4~5.
5: Send the domain-specific features and cluster centroids to the server.
6: **end for**
7: Generate pseudo samples on the server using Eq.7.
8: Finetune the classification model $h = F_\theta \circ E_\theta$ using the pseudo samples.

of the whole federated learning framework is:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_k}\left[l_k(\mathbf{w};\mathbf{x})\right] \tag{1}$$

where l_k is the local objective function for the k-th client, $\mathbf{w}$ is the parameters of the aggregated model.

To reduce communication and computation costs and make it suitable for real-world scenarios, such as the devices in autonomous driving, we impose two constraints in this setting: 1) Clients cannot conduct model training and can only conduct model inference. 2) The federated training process can only involve one round of communication.

3.2 FedDISC

Our method can be divided into four steps: prototype extraction, pseudo labeling, feature processing, and image generation.

Prototype Extraction. The first step is prototype extraction. We utilize a pre-trained CLIP image encoder E_θ to extract the features of all labeled data on the server. We assume that the server contains all possible categories that may appear on the clients, but each category on the server has a relatively uniform distribution.

After obtaining the features of the labeled images on the server, we extract prototypes p_j of all categories by calculating the average of all features with the same category:

$$p_j = \frac{\sum_{(\mathbf{x}_i^s, y_i)\in\mathcal{D}_s} E_\theta(\mathbf{x}_i^s) * \mathbb{I}(y_i = j)}{\sum_{(\mathbf{x}_i^s, y_i)\in\mathcal{D}_s} \mathbb{I}(y_i = j)} \tag{2}$$

,where $\mathbb{I}$ is the indicator function. Finally, we send the extracted category prototypes p_j and the pre-trained CLIP image encoder E_θ to all the clients.

Pseudo Labeling. The second step is pseudo labeling. For client k, after receiving the encoder E_θ and the prototypes p_j from the server, the client uses E_θ to extract features of all unlabeled images in $\mathcal{D}_k$ and calculates the similarities between each feature $E_\theta(\mathbf{x}_i^k)$ and all category prototypes p_j:

$$sim(E_\theta(\mathbf{x}_i^k), p_j) = \frac{E_\theta(\mathbf{x}_i^k)^\top p_j}{\left\|E_\theta(\mathbf{x}_i^k)\right\|\left\|p_j\right\|}, \mathbf{x}_i^k \in \mathcal{D}_k, j = 1, ..., M \tag{3}$$

Based on the similarities, each image $\mathbf{x}_i^k$ is assigned with a pseudo-label $\hat{y}_i^k$, where $\hat{y}_i^k = \arg\max_j sim(E_\theta(\mathbf{x}_i^k), p_j)$. Due to the distribution differences between $\mathcal{D}_s$ and $\mathcal{D}_k$, there is a possibility of making mistakes in pseudo labeling. In traditional semi-supervised federated learning methods, pseudo-labels are directly used for self-training. Therefore, various semi-supervised federated learning methods are required to improve the quality of pseudo-labels. However, in our method, the text prompts of the labels are added as auxiliary information during conditional generation on the server, and the adverse influence of the mistakes in pseudo labeling will be greatly reduced.

Feature Processing. The third step is feature processing. After obtaining the pairs of the unlabeled client data and their pseudo labels $\{\mathbf{x}_i^k, \hat{y}_i^k\}_{i=1}^{N_k}$, we cannot upload all the client features to the server due to the communication considerations. Therefore, taking category j as an example, we cluster the client features belonging to the category j and select L representative cluster centroids $\{\mathbf{z}_{j,l}^k\}_{l=1}^{L}$ to upload. The objective of the clustering process is as follows:

$$\arg\min_{\mathbf{z}_{j,l}^k} \sum_{l=1}^{L} \sum_{\mathbf{x}_i^k\in\mathcal{D}_k} \left\|\mathbf{x}_i^k - \mathbf{z}_{j,l}^k\right\|^2 * \mathbb{I}\left(\hat{y}_i^k = j\right) \tag{4}$$

Compared with randomly selecting L features for uploading, the personalized distribution information and semantic information contained in the cluster centroids are clearer. Since only a small number of features are selected and each feature will generate multiple images on the server, the quality of the uploaded features is crucial.

Meanwhile, we obtain the domain-specific features $\{\mathbf{g}_j^k\}_{j=1}^{M}$ for each category on client k by averaging all the features belonging to the same category. We weaken the individuality of each image and highlight the commonality of each category on the clients in the domain-specific features. During the conditional generation on the server, we can generate images that complied to different client distributions by randomly combining the cluster centroids with different domain-specific features.

After computing the cluster centroids and the domain-specific features, for privacy protection, we add noise to all these features. The noise-adding process is as follows:

$$\bar{\mathbf{z}}_{j,l}^k = \sqrt{\alpha_n}\mathbf{z}_{j,l}^k + \sqrt{1-\alpha_n}\epsilon, \bar{\mathbf{g}}_j^k = \sqrt{\alpha_n}\mathbf{g}_j^k + \sqrt{1-\alpha_n}\epsilon, \tag{5}$$

, where $\epsilon \sim \mathcal{N}(0, \mathcal{I})$, n is a hyperparameter controlling the intensity of the noise, and $n \in \{0, \ldots, T\}$. We follow the noise-adding process in Stable Diffusion [33] and perform a noise-adding process with a specific timestep to these image features.

Assuming that an attacker can obtain the uploaded $\mathbf{z}$ in each round, which can be seen as a query. Through the Gaussian mechanism in Eq 5, we can achieve differential privacy [1] on each query. It can protect private information from being leaked through multiple queries. And we will further discuss in section 4 and the supplementary materials that using cluster centroids as conditions, Stable Diffusion cannot restore the privacy-sensitive information of the original image, such as faces, text, etc. After this step, cluster centroids $\{\bar{\mathbf{z}}_{j,l}^k\}_{l=1}^{L}, j = \{1, \ldots, M\}$ and domain-specific features $\{\bar{\mathbf{g}}_j^k\}_{j=1}^{M}$ are uploaded for the conditional generation on the server.

Image Generation. After receiving the cluster centroids $\bar{\mathbf{z}}_{j,l}^k$ and the domain-specific features $\bar{\mathbf{g}}_j^k$ uploaded from the clients, for each cluster centroid $\bar{\mathbf{z}}_{j,l}^k$, the server will randomly combine $\bar{\mathbf{z}}_{j,l}^k$ with the domain-specific features which have the same pseudo-label j. The selected domain-specific features $G_{j,l}^k$ are as follows:

$$G_{j,l}^k = \{\bar{\mathbf{g}}_j^{k_0}, \ldots, \bar{\mathbf{g}}_j^{k_R}\}, k_0, \ldots, k_R \in \lfloor \text{uniform}(0, K+1) \rfloor \quad (6)$$

By combining domain-specific features from different clients in this manner, we generate images that complied to different client distributions from a single cluster centroid. In addition, with the help of text prompts of the pseudo-labels C_j, $j \in \{0, \ldots, M\}$, the generated images have the correct semantic information corresponding to the given pseudo-labels.

As for the generating process, since we aim to use the cluster centroids $\bar{\mathbf{z}}_{j,l}^k$, domain-specific features $\bar{\mathbf{g}}_j^{k_i}$, and the text prompts of categories C_j as conditions for generation, the conditional probability distribution of the generating process can be written in the following form:

$$p(x|\bar{\mathbf{z}}_{j,l}^k, \bar{\mathbf{g}}_j^{k_i}, C_j) \propto p(x|C_j) p(\bar{\mathbf{z}}_{j,l}^k|x, C_j) p(\bar{\mathbf{g}}_j^{k_i}|x, C_j)$$

Since the given noise x is initially sampled from a Gaussian distribution, independent of the used cluster centroids and domain-specific features, the above formula can be rewritten as follows:

$$p(x|\bar{\mathbf{z}}_{j,l}^k, \bar{\mathbf{g}}_j^{k_i}, C_j) \propto p(x|C_j) \frac{p(x|\bar{\mathbf{z}}_{j,l}^k, C_j)}{p(x|C_j)} \frac{p(x|\bar{\mathbf{g}}_j^{k_i}, C_j)}{p(x|C_j)}$$

Therefore, specifically, we input the feature of category prompt C_j with a cluster centroid $\bar{\mathbf{z}}_{c,l}^k$, a domain-specific feature $\bar{\mathbf{g}}_c^{k_i}$, and without any image feature to the pre-trained diffusion model to obtain three predicted noises. And we accumulate these three predicted noises in the following formula to obtain the final predicted noise:

$$\hat{\epsilon}_\theta(\mathbf{x}_t, t|\bar{\mathbf{z}}_{j,l}^k, \bar{\mathbf{g}}_c^{k_i}, C_j) = \epsilon_\theta(\mathbf{x}_t, t|C_j) + w_f(\epsilon_\theta(\mathbf{x}_t, t|\bar{\mathbf{z}}_{jc,l}^k, C_j) - \epsilon_\theta(\mathbf{x}_t, t|C_j)) + w_g(\epsilon_\theta(\mathbf{x}_t, t|\bar{\mathbf{g}}_j^{k_i}, C_j) - \epsilon_\theta(\mathbf{x}_t, t|C_j))$$

, where w_f and w_g are the weights of the predicted noises. Overall, the generated images are obtained through the denoising process:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\Big(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\hat{\epsilon}_\theta(\mathbf{x}_t, t|\bar{\mathbf{z}}_{j,l}^k, \bar{\mathbf{g}}_j^{k_i}, C_j)}{\sqrt{\alpha_t}}\Big) + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \hat{\epsilon}_\theta(\mathbf{x}_t, t|\bar{\mathbf{z}}_{j,l}^k, \bar{\mathbf{g}}_j^{k_i}, C_j) + \sigma_t \varepsilon_t, t = T, \ldots, 1 \quad (7)$$

After obtaining the generated images, since both the cluster centroids and domain-specific features used for image generation have their corresponding pseudo-labels, the generated images are pseudo-labeled. So we can directly fine-tune a classification model h with the generated dataset for downstream classification tasks. The classification model $h = F_\theta \circ E_\theta$ is a composite of the pre-trained CLIP image encoder E_θ and a linear classifier F_θ.

4 EXPERIMENTS

We aim to answer the following questions:

- Can diffusion models' powerful image generation capability liberates the client training in traditional federated learning?
- Can high-quality images that complied to the client distributions be generated on the server by conditioning on the uploaded features?
- Whether the generated images contain any privacy-sensitive information of the clients or not?
- How does the performance of our method compare with other SOTA methods?

We validate the performance of our method on three large-scale image classification datasets. For the non-iid problem, we consider two scenarios: 1) Based on the DomainNet dataset, we conduct experiments on five clients with the same categories but different image styles. 2) We set up five clients on the OpenImage and NICO++ datasets, each containing different subcategories belonging to the same super-category. We also discuss the influences of various hyperparameters and the privacy issues of our work.

4.1 Experimental Setup

Dataset. We use three datasets to evaluate the performance of FedDISC: DomainNet [29], OpenImage [20], and NICO++ [48]. For all these datasets, the resolutions of the input images for feature extraction and classification are 224x224 and the resolutions of generated images are 768x768. DomainNet is a large-scale dataset consisting of six domains, with each domain containing 345 categories. We set the images from domain *real* as the labeled data on the server, and the images from the other five domains as the unlabeled data on five clients. OpenImage has a hierarchy of categories, from which we select 20 super-categories with 6 subcategories in each super-category. These 6 subcategories are used as the server and five clients. Each subcategory contains 300-500 images, and the specific selection of categories can be found in the supplementary material. NICO++ can be divided into a common context dataset (NICO++_C) and a unique context (NICO++_U), both containing 60 classes. NICO++_C is divided into six fixed domains (*rock, autumn, water, outdoor, grass, dim*), with each domain containing about 3k-4k images. In NICO++_U, each class has six unique domains, such as *"bear"* being split into *"black, in cage, lying, roaring, sitting, wombat"*, with each domain containing about 7k-16k images. In OpenImage, NICO++_C and NICO++_U, we randomly select one domain as the server and the other five domains as the client.

As for the text prompts to be input, in order to simulate realistic scenarios, we avoid using the descriptions of the styles or subcategories as the conditions. In the DomainNet dataset, we only use category labels as prompts, such as *"airplane"* or *"bridge"*, without using the descriptions of styles like *"clipart"* or *"quickdraw"*. Similarly, in the OpenImage and NICO++, we only use the super-categories as the text prompts without specifying subcategories. For example, in the NICO++_U dataset, the *"flower"* class contains six domains, including *"bouquet"* and *"wreath"*, but we only use *"flower"* as the text prompt, without providing the specific subcategory.

Baselines. The model architecture we used is a composite of a pre-trained CLIP and a linear classifier. During fine-tuning, we freeze the pre-trained CLIP and only fine-tune the additional classifier. We mainly compare our method with 5 baseline approaches: 1) **Fine-tuning**: directly fine-tuning the model with the uploaded cluster centroids and corresponding pseudo-labels. 2) **CLIP Zero-shot**: using the zero-shot classification capability of pre-trained CLIP

Table 1: The overall performance in different datasets.

		DomainNet						OpenImage					
	Rounds	clipart	infograph	painting	quickdraw	sketch	average	client0	client1	client2	client3	client4	average
Fine-tuning	1	67.57	45.47	65.28	10.42	62.14	50.18	36.67	46.81	45.43	47.17	42.10	43.64
FedAvg-sm	30	49.95	30.67	51.07	1.74	38.46	34.38	41.11	44.06	46.57	47.45	37.63	43.36
CLIP Zero-shot	1	65.86	40.50	62.25	13.36	57.92	47.98	56.00	40.61	40.28	44.06	61.45	48.48
SemiFL	500	69.55	47.16	64.54	7.02	63.32	50.32	48.15	52.78	61.05	55.23	46.16	52.67
RSCFed	100	71.50	45.73	61.96	11.53	65.03	51.15	28.97	38.04	40.82	33.98	36.35	35.63
FedDISC	1	70.48	41.71	66.84	15.90	63.37	**51.66**	56.11	62.49	62.53	59.16	56.77	**59.41**
		NICO++_C						NICO++_U					
		dim	grass	outdoor	rock	water	average	client0	client1	client2	client3	client4	average
Fine-tuning	1	86.50	89.39	83.61	87.21	76.95	84.73	84.75	79.08	81.48	86.58	83.52	83.08
FedAvg-sm	30	86.98	90.82	82.68	87.57	74.48	84.51	83.26	73.30	77.93	80.80	79.28	78.91
CLIP Zero-shot	1	78.66	85.26	80.01	80.70	72.14	79.35	89.20	89.24	87.19	85.50	88.60	87.94
SemiFL	500	87.55	89.27	81.93	87.16	77.01	84.58	78.21	74.70	79.87	80.69	77.02	78.10
RSCFed	100	52.08	60.15	52.60	55.35	43.89	52.81	71.88	64.14	70.82	69.71	69.67	69.24
FedDISC	1	87.18	91.09	86.36	87.66	77.42	**85.94**	90.09	89.74	88.15	91.90	88.98	**89.77**

without any fine-tuning. 3)**FedAvg-sm**: Semi-supervised FedAvg, using clients' data and corresponding pseudo-labels for conducting FedAvg, which requires client training and multiple rounds of communications. In addition, we evaluate the SOTA semi-supervised federated learning methods: 4)**SemiFL** [5] and 5)**RSCFed** [23]. Noted that as there is currently no one-shot semi-supervised federated learning method, the chosen semi-supervised methods require multiple rounds of communication for comparison.

Implementation Details. Our method is mainly implemented based on *PyTorch* and *Diffusers*. For the base model, we use the *Stable-Diffusion-Unclip-v2.1* [33] from *HuggingFace*, where the pre-trained diffusion model is used as ϵ_θ for image generation, and the pre-trained CLIP [31] from *HuggingFace* is used as E_θ for image encoding. During the image generating process, we use the DDIM scheduler and set the number of iterations for denoising to 20. Unless otherwise specified, the number of uploaded cluster centroids L is set to 5. Conditioned on each cluster centroid, the number of generated images R is 10. The weights of noises w_f and w_g are 2. The intensity of the noise-adding process is 200. During the subsequent fine-tuning phase, we use the SGD optimizer with a learning rate of 0.01 and a batch size of 256. All experiments are completed with four NVIDIA GeForce RTX 3090 GPUs.

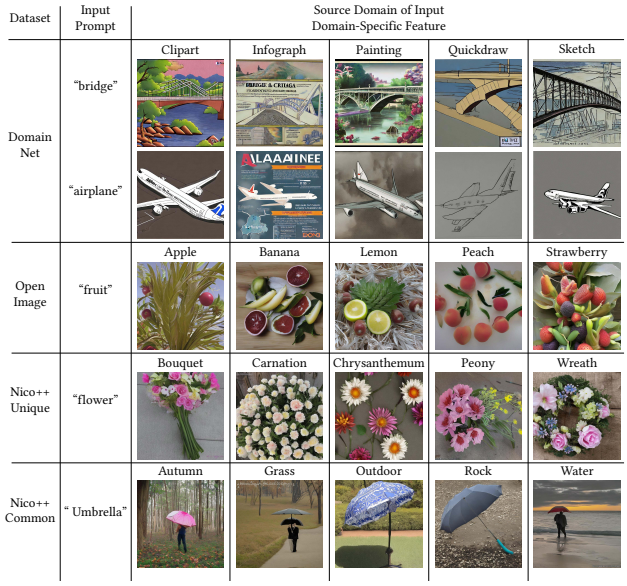

Figure 3: Generated images with different styles.

4.2 Main Results

In Table 1, we present the performances of our method and various baselines on four datasets: DomainNet, OpenImage, NICO++_C, and NICO++_U. We highlight several phenomena:

- Despite the powerful generalization ability of CLIP, directly using the pre-trained CLIP to perform zero-shot classification still cannot achieve the best performance in all clients, and therefore further fine-tuning is needed.
- When directly fine-tuning with uploaded features, there is still potential for improvement in the final classification performance due to the noise in the uploaded cluster centroids and the problem of data diversity.
- Existing federated semi-supervised methods require a large number of communication rounds and cannot guarantee stable performance improvements. However, FedDISC achieves the best performance with only one communication round. This demonstrates the enormous potential of pre-trained diffusion models in the federated learning scenarios.
- From the results on DomainNet, we can see that FedDISC has good performance on all clients except *infograph*. We think the reason is that there is a lot of textual information in this domain, but currently, the Stable Diffusion still has limited support for text present in the images. We think with the upgrading of Stable Diffusion, the performance of FedDISC will also improve.
- On the OpenImage, NICO++_C, and NICO++_U datasets, compared with other baselines, FedDISC exhibits a significant performance improvement. We believe that there are two reasons for this: 1) The Stable Diffusion is exposed to more realistic images during pre-training, and the generation quality of the realistic images is better. 2) In the process of conditional generation, we

Table 2: The influence of the number of cluster centroids.

	client0	client1	client2	client3	client4	average
L=3 Fine-tuning	36.01	46.01	44.59	45.55	41.87	42.81
L=3 FedDISC	56.33	61.93	58.62	56.71	58.74	**58.47**
L=5 Fine-tuning	36.67	46.81	45.43	47.17	42.10	43.64
L=5 FedDISC	56.11	62.49	62.53	59.16	56.77	**59.41**
L=10 Fine-tuning	37.55	45.86	44.85	46.01	42.15	43.28
L=10 FedDISC	57.16	63.84	61.12	57.91	59.13	**59.83**

Table 3: The influence of the number of generated images.

	client0	client1	client2	client3	client4	average
R=3	53.41	62.15	61.31	56.87	55.49	57.85
R=5	54.58	63.47	61.26	58.19	57.57	59.01
R=10	56.11	62.49	62.53	59.16	56.77	**59.41**

Table 4: The influence of different conditions.

domain-specific features	cluster centroids	client0	client1	client2	client3	client4	average
✓		67.79	40.02	63.59	13.77	60.57	49.15
	✓	65.83	38.27	64.56	14.30	60.37	48.66
✓	✓	70.48	41.71	66.84	15.90	63.37	**51.66**

only input the prompt of the super-categories, which less limits the model and generates more diverse images.

- Compared with other baseline methods, the performance of FedDISC is relatively stable on each dataset. Other baseline methods are more or less influenced by the factors such as the quality of the client images and the accuracy of pseudo-labels. However, FedDISC uses both image features and text prompts as conditions to generate pseudo-samples for fine-tuning. Even if the features of the clients are suboptimal or the pseudo-labels are incorrect, high-quality images with expected semantic information can be generated, which increases the adaptability of the method in complex realistic scenarios.

At the same time, from Figure 3 and the other visualization results in the supplementary materials, it can be seen that on DomainNet, our method can indeed generate high-quality images that complied to various client styles while being semantically correct, and on OpenImage, NICO++_C, and NICO++_U, our method can generate images that complied to the client subcategories, which means the generated images can introduce new knowledge for the classification model during fine-tuning.

4.3 Discussions

We conduct four kinds of experiments in DomainNet and OpenImage to discuss the influences of various hyperparameters, including the variations in the number of uploaded cluster centroids L, the number of generated images per cluster centroid R, the use of domain-specific features and cluster centroids, and the number of categories M. We also conduct some visualization experiments on DomainNet to discuss the privacy issues of FedDISC.

The number of uploaded cluster centroids. Firstly, we perform experiments on the OpenImage dataset to discuss the influence

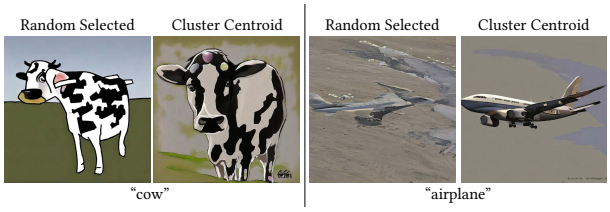

Figure 4: With/Without cluster centroids.

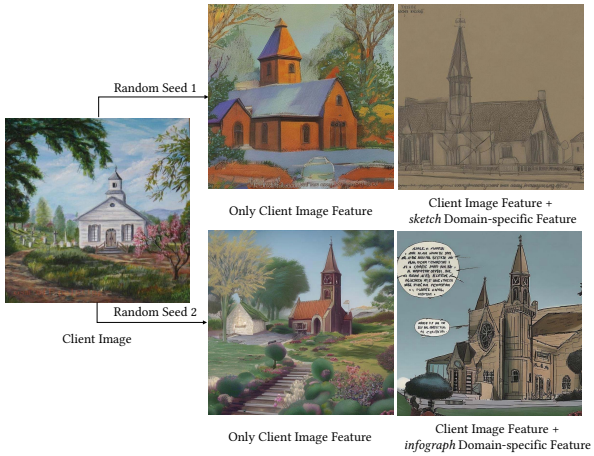

Figure 5: With/Without domain-specific features.

of the number of the uploaded cluster centroids L. Since this number is related to the performance of Fine-tuning, we also test the performance of Fine-tuning under different L for comparison. From Table 2, three observations can be shown: 1) FedDISC outperforms Fine-tuning in all cases; 2) as L increases, the performance of both methods gradually improves, which is reasonable, because increasing L represents the increment in data diversity; 3) when L increases from 3 to 5, the performance improvement is greater than when it increases from 5 to 10, and even the performance of Fine-tuning declines on some clients under the latter case. We believe the reason is that the uploaded features are clustered. In most cases, uploading a small number of cluster centroids is already sufficient to represent the semantics of the subcategories. Increasing the number of uploaded cluster centroids may cover some outliers, which may have adverse influence on fine-tuning and generating.

The number of generated images. We discuss the number of images generated by each cluster centroid on the OpenImage dataset and find two key points from Table 3: 1) the overall performance of the method gradually improves as R increases. This is reasonable as the increment of R indicates the increment in the amount of training data during the subsequent fine-tuning process. 2) when R increases from 3 to 5, the performance improvement is greater compared to when R increases from 5 to 10, despite the latter involving a greater change in the number. This is similar to the third point in our previous discussion on L. In most cases, generating a small number of images is sufficient since the complexity of the distributions of each category is limited.

Table 5: The influence of the number of categories in the dataset.

	DomainNet 90 class				DomainNet 150 class				DomainNet 345 class			
	Fine-tuning	FedAvg	Zero-shot	FedDISC	Fine-tuning	FedAvg	Zero-shot	FedDISC	Fine-tuning	FedAvg	Zero-shot	FedDISC
client0	79.61	80.44	79.28	84.79	73.14	71.57	76.26	80.49	67.57	49.95	65.86	70.48
client1	62.40	62.08	57.79	68.13	55.16	51.70	45.84	56.67	45.47	30.67	40.50	41.71
client2	75.72	72.88	72.45	76.00	69.74	66.62	69.43	72.41	65.28	51.07	62.25	66.84
client3	21.70	5.54	22.50	28.17	15.56	5.96	19.57	21.65	10.42	1.74	13.36	15.90
client4	75.26	72.98	71.67	80.72	67.82	64.52	67.46	75.48	62.14	38.46	57.92	63.37
average	62.94	58.784	60.74	**67.56**	56.28	52.07	55.71	**61.34**	50.17	34.38	47.98	**51.66**

Domain-specific features and cluster centroids. The third part concerns the discussion on the roles of domain-specific features and cluster centroids. As mentioned in Section 3, we believe that adding domain-specific features can enhance the domain-specific characteristics of the categories on the clients and controllably generate images that complied to different client distributions for a single cluster centroid. The cluster centroids can avoid uploading features from semantically ambiguous images.

Compared to other datasets, DomainNet has more distinct differences between the clients and more cases of semantically ambiguous images. Therefore, in this part, we mainly conduct our experiments on DomainNet. We compare FedDISC with cases where domain-specific features are not used during generation, and cases where cluster centroids are not uploaded, but an equal number of client features are randomly selected and uploaded. As shown in Table 4, the removal of either domain-specific features or cluster centroids has a significant influence on the performance of FedDISC.

The visualization results in Figure 4 demonstrate that without the cluster centroids, the semantic information of the generated images becomes ambiguous and may lead to generating images that do not match the given text prompts. Figure 5 shows that removing the domain-specific features leads to the style of the generated images being monotonous, which reduces the diversity of the generated images. These results are consistent with our goals of using domain-specific features and cluster centroids.

It should be noted that in Figure 4, the two images used for comparison have the same factors that may influence the generation, such as the domain-specific features, the random noises sampled at each timestep, etc. It is similar in Figure 5 with only the domain-specific features being different. We can find evidence in these figures. For example, the left images of Figure 4 have same domain-specific features from the *painting* client, and the right images of Figure 4 show a very similar shape and color on the right part of both images, which indicates that they are generated from identical initial noise. In Figure 5, the overall structures of the churches also demonstrate that the initial noises of the two images of each row are identical. These results further demonstrate the effectiveness of using cluster centroids and domain-specific features.

The number of categories in the dataset. Since DomainNet has the largest number of categories, we conduct experiments on this dataset by selecting the first 90 categories, 150 categories, and all categories for comparison. As shown in Table 5, as the number of image categories decreases, the performance improvement of FedDISC becomes more significant. We believe this is because, during the generating process, some semantically ambiguous images may still be generated. The reasons behind this are multifaceted, including poor quality of the selected cluster centroids and limitations in the generation capability of Stable Diffusion. These images have less influence when the number of categories is small since the classification task is relatively straightforward. As the number of categories increases, the influence of these images become more significant, finally adversely influence the performance of FedDISC.

Raw Client Image | Generated Image | Raw Client Image | Generated Image

Figure 6: The comparison between the raw client images and their corresponding generated images.

The privacy issues. Although we can achieve differential privacy by adding noises, there is still a risk of privacy-sensitive information leakage due to the generation of the images complied to the distributions of the clients. In real-world scenarios, privacy issues are subjective. To demonstrate whether the generated images in FedDISC contain any privacy-sensitive information of the clients, we conduct some visualization experiments comparing the raw client images and the corresponding generated images. As shown in Figure 6, the generated images do not contain any privacy-sensitive information from the clients. Due to space limitations, we provide further discussion and analysis on privacy issues through additional visualization experiments in the supplementary materials.

5 CONCLUSION

In this paper, we explore the task of one-shot semi-supervised federated learning and propose a new approach that integrates pre-trained diffusion models into the federated learning framework for the first time. Unlike traditional semi-supervised federated learning, our method requires only one round of communication and does not require client backpropagation training. By leveraging the powerful conditional image generation capabilities of pre-trained diffusion models, our approach achieves SOTA performance compared with existing multi-round semi-supervised federated learning methods. Extensive experiments demonstrate the significant potential of pre-trained diffusion models in federated learning.

REFERENCES

[1] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part I*. Springer, 635–658.

[2] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[3] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. 2022. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12413–12422.

[4] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 8780–8794.

[5] Enmao Diao, Jie Ding, and Vahid Tarokh. 2021. SemiFL: Communication efficient semi-supervised federated learning with unlabeled clients. *arXiv preprint arXiv:2106.01432* 3 (2021).

[6] Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. 2020. Personalized Federated Learning with Moreau Envelopes. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/f4f1f13c8289ac1b1ee0ff176b56fc60-Abstract.html

[7] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. 2023. Reduce, Reuse, Recycle: Compositional Generation with Energy-Based Diffusion Models and MCMC. *arXiv preprint arXiv:2302.11552* (2023).

[8] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems* 33 (2020).

[9] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. 2022. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10696–10706.

[10] Tao Guo, Song Guo, Junxiao Wang, and Wenchao Xu. 2022. PromptFL: Let Federated Participants Cooperatively Learn Prompts Instead of Models–Federated Learning in Age of Foundation Model. *arXiv preprint arXiv:2208.11625* (2022).

[11] Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. [n.d.]. Data-Free One-Shot Federated Learning Under Very High Statistical Heterogeneity. In *The Eleventh International Conference on Learning Representations*.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

[13] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. 2023. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778* (2023).

[14] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. 2021. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7865–7873.

[15] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. 2021. Federated Semi-Supervised Learning with Inter-Client Consistency & Disjoint Learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=ce6CFXBh30h

[16] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*. PMLR, 5132–5143.

[17] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364* (2022).

[18] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. 2022. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2426–2435.

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643* (2023).

[20] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. 2020. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision* 128, 7 (2020), 1956–1981.

[21] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks.. In *MLSys*, Inderjit S. Dhillon, Dimitris S. Papailiopoulos, and Vivienne Sze (Eds.). mlsys.org. http://dblp.uni-trier.de/db/conf/mlsys/mlsys2020.html#LiSZSTS20

[22] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189* (2019).

[23] Xiaoxiao Liang, Yiqun Lin, Huazhu Fu, Lei Zhu, and Xiaomeng Li. 2022. RSCFed: random sampling consensus federated semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10154–10163.

[24] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. 2022. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778* (2022).

[25] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. 2022. Compositional visual generation with composable diffusion models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*. Springer, 423–439.

[26] Zhaoyang Lyu, Xudong Xu, Ceyuan Yang, Dahua Lin, and Bo Dai. 2022. Accelerating diffusion models via early stop of the diffusion process. *arXiv preprint arXiv:2205.12524* (2022).

[27] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[28] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).

[29] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1406–1415.

[30] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. 2022. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10619–10629.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[32] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295* (2020).

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

[35] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. 2022. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–10.

[36] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* 35 (2022), 36479–36494.

[37] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. PMLR, 2256–2265.

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).

[39] Shangchao Su, Bin Li, and Xiangyang Xue. 2023. One-shot Federated Learning without server-side training. *Neural Networks* (2023). https://doi.org/10.1016/j.neunet.2023.04.035

[40] Shangchao Su, Mingzhao Yang, Bin Li, and Xiangyang Xue. 2022. Cross-domain Federated Adaptive Prompt Tuning for CLIP. *arXiv preprint arXiv:2211.07864* (2022).

[41] Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. 2022. Dual diffusion implicit bridges for image-to-image translation. In *The Eleventh International Conference on Learning Representations*.

[42] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems 33* (2020).

[43] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. 2022. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952* (2022).

[44] Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. 2022. Dense: Data-free one-shot federated learning. *Advances in Neural Information Processing Systems* 35 (2022), 21414–21428.

[45] Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543* (2023).

[46] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. 2021. Personalized Federated Learning with First Order Model Optimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=ehJqJQk9cw

[47] Qinsheng Zhang and Yongxin Chen. 2022. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902* (2022).

[48] Xingxuan Zhang, Linjun Zhou, Renzhe Xu, Peng Cui, Zheyan Shen, and Haoxin Liu. 2022. Nico++: Towards better benchmarking for domain generalization. *arXiv preprint arXiv:2204.08040* (2022).

[49] Zhengming Zhang, Yaoqing Yang, Zhewei Yao, Yujun Yan, Joseph E Gonzalez, Kannan Ramchandran, and Michael W Mahoney. 2021. Improving semi-supervised federated learning by reducing the gradient diversity of models. In *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 1214–1225.