Navigating Heterogeneity and Privacy in One-Shot Federated Learning with Diffusion Models

Matias Mendieta, Guangyu Sun, and Chen Chen

Center for Research in Computer Vision, University of Central Florida, FL, USA
{matias.mendieta, guangyu.sun}@ucf.edu; chen.chen@crcv.ucf.edu

Abstract. Federated learning (FL) enables multiple clients to train models collectively while preserving data privacy. However, FL faces challenges in terms of communication cost and data heterogeneity. One-shot federated learning has emerged as a solution by reducing communication rounds, improving efficiency, and providing better security against eavesdropping attacks. Nevertheless, data heterogeneity remains a significant challenge, impacting performance. This work explores the effectiveness of diffusion models in one-shot FL, demonstrating their applicability in addressing data heterogeneity and improving FL performance. Additionally, we investigate the utility of our diffusion model approach, FedDiff, compared to other one-shot FL methods under differential privacy (DP). Furthermore, to improve generated sample quality under DP settings, we propose a pragmatic Fourier Magnitude Filtering (FMF) method, enhancing the effectiveness of generated data for global model training.

1 Introduction

Federated learning (FL) is a distributed machine learning technique that enables multiple clients to participate in the training process in a privacy-preserving manner. In FL, each client trains a local model on its own data and sends the model to a central server. The server combines these updates to improve the global model, which is then sent back to the clients. This approach ensures that the clients' data is kept private while enabling the central server to learn from the collective knowledge of all participating users [16].

However, FL poses significant challenges in terms of communication cost and data heterogeneity across clients. Communication cost is a major bottleneck in FL systems, as clients need to communicate frequently with the server over multiple rounds during the training process [3, 16, 28]. This leads to a high communication overhead, making the process slow or simply infeasible. To overcome this challenge, **one-shot federated learning** has recently gained traction in the research community [10, 29, 35, 36]. In this setting, clients only communicate once with the server during the training process, significantly reducing the communication requirements. This approach not only improves the efficiency of the training process but also provides a better framework for privacy and application. Specifically, one-shot FL provides better security against eavesdropping

attacks, where adversaries attempt to steal or tamper with the information being sent between clients and the server [22]. By only requiring one round of communication, one-shot FL significantly reduces the likelihood of such attacks. Furthermore, traditional multi-round training may not be a practical option in some cases, such as that of model markets [20]. In these scenarios, models are trained to convergence by a participating user, and simply made available as a pretrained model to potential buyers, without any option for iterative communication.

However, another significant challenge in federated learning still remains, and that is, the data heterogeneity problem [16, 17, 21, 25]. In FL, clients often have very different data distributions, making optimization particularly challenging across the federated system. In the one-shot setting, this is especially detrimental to performance. Without the luxury of multiple communication rounds, the resulting models will be significantly biased towards their narrow data distribution and difficult to reconcile into a global model. Knowledge distillation-based approaches have been studied in the literature in an attempt to address these problems [10, 20, 35]. Nonetheless, these methods still struggle immensely under high heterogeneity, resulting in large drops in performance.

Yet, another class of model is potentially well-suited for such heterogeneous distributions at the clients. Rather than simply employing discriminative models to train on the clients, one could instead leverage generative models. These generative models can then be gathered from the clients and inferenced on the server to form a dataset for global model training, eliminating the need for the challenging reconciliation process required for discriminative models. [13] conducted a preliminary study of such a framework with conditional variational autoencoders (CVAEs) [30] for one-shot FL, but there is still much to investigate in this paradigm. *Specifically, we consider two primary research questions (RQ) in this work.*

RQ1. First, we explore the utility of diffusion models in federated learning and their potential for improving the performance of the one-shot FL process. Diffusion models [14] have recently emerged as prominent approaches for image generation, inspiring our investigation. We suggest that specific traits of diffusion models could provide advantages for one-shot FL, as discussed in Section 3. We then validate this hypothesis through comprehensive experiments with our approach, FedDiff, across various settings.

RQ2. Second, we investigate one-shot FL methods under provable privacy budgets with differential privacy (DP), as this aspect is not addressed by existing state-of-the-art (SOTA) one-shot FL works. Safeguarding model privacy is critical in this setting, as the client models obtained in one-shot FL can be reused multiple times or even traded in a model market. Furthermore, in light of recent work [5], we examine the potential memorization of diffusion models within our FedDiff approach and the effectiveness of DP as a mitigation strategy.

After studying these research questions, we further explore a simple technique for improving the performance of our FedDiff method under DP settings. We observe that the quality of generated samples may deteriorate under DP con-

straints, rendering some samples counterproductive to the training of the global model. To improve the quality and consistency of the synthetic data, we propose a straightforward filtering approach, termed Fourier Magnitude Filtering (FMF). FMF leverages sample magnitudes derived from the Fourier transform to guide the selection of valuable samples. The resulting filtered dataset substantially improves the utility of the generated data, particularly in challenging conditions, as detailed in Section 5.3. Therefore, in this work, our **contributions** are summarized as follows:

- We contribute to the FL literature with the first study exploring diffusion models in one-shot federated learning. Our comprehensive investigation unveils the unique advantages inherent to diffusion models, which enhances the overall performance of one-shot FL while also addressing the significant challenges of data heterogeneity. We therefore establish a novel approach, FedDiff, that not only ensures superior model performance but also aligns with the core requirements of one-shot FL.
- We further study the privacy and utility of both discriminative and generative-based SOTA one-shot FL methods with DP guarantees under heterogeneous settings. We find that our FedDiff approach outperforms all other methods by a significant margin (from $\sim$5% to $\sim$20% across many datasets and settings), even when differential privacy is employed.
- While FedDiff performs very well, we note that sample quality is affected under DP. Therefore, to improve performance in such conditions, we propose a simple Fourier Magnitude Filtering (FMF) approach, which improves the effectiveness of the generated data for global model training by removing low-quality samples.

2 Background and Preliminaries

2.1 One-shot Federated Learning

Federated learning (FL) has emerged as a promising paradigm for collaborative machine learning across decentralized devices while preserving data privacy. The seminal work by McMahan et al. [24] introduced the concept of FL, where model updates are computed locally on user devices and aggregated on a central server. However, in the standard FL process, many iterative communication rounds are required for convergence. One-shot FL, therefore, studies how to effectively learn in this distributed setting in a single round, thereby mitigating the need for many communication rounds. Several approaches have been proposed to tackle the unique characteristics of one-shot FL. [10] introduce the one-shot federated learning framework and study several baseline approaches. In [10] and [20], distillation approaches are studied using the ensemble of client models to the global model, and assume a public dataset for this purpose. However, such an assumption is limited, as public data related to the domain of interest is often not available. A data-free method within the distillation methodology was proposed by [35], where a generative adversarial network (GAN) is trained

at the server level to generate the data for distillation, and iteratively optimized between distilling to the server model and training the GAN with the ensemble of client models.

Nonetheless, these methods still struggle with heterogeneous environments, as we find in Section 3. Generative models on the client are well-suited for better undertaking in such settings, as they can focus on the narrow client distributions and simply generate data at the central location. [13] introduce the use of CVAEs in highly heterogeneous one-shot FL. However, CVAEs exhibit suboptimal sample quality, a limitation that becomes markedly exacerbated with more complex datasets and when subjected to the constraints of DP, which are not explicitly addressed in the study by [13]. In this work, we investigate diffusion models in one-shot FL and leverage their unique characteristics for the task, illustrating their potential in a variety of difficult FL settings and privacy guarantees.

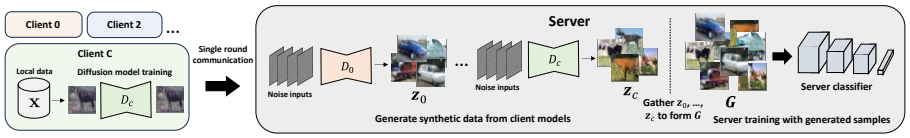


Fig. 1: Our one-shot FL approach, FedDiff. We first train a class-conditioned diffusion model on local data $\mathbf{x}$ at the clients. After completing training, the local diffusion models D_0, D_1, ..., D_c are gathered by the server, where they are used to generate data $\mathbf{z}_0$, $\mathbf{z}_0$, ..., $\mathbf{z}_c$, which are combined to form the global training data $\mathbf{G}$. The global model is then trained on this synthetic dataset $\mathbf{G}$.

2.2 Diffusion Probabilistic Models

Diffusion probabilistic models [6,14], or simply diffusion models as they are now commonly referenced (DM), have gained traction for application in generative vision tasks. Simply put, DMs aim to learn the backward process that can iteratively denoise an image corrupted with Gaussian noise back to the original. Specifically, as detailed in [14], noise is introduced to a given sample via a Markovian chain forward process

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \tag{1}$$

where T is the total number of iterations (or timesteps) applied, and $q(x_t|x_{t-1})$ is parameterized by $\mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$. β is a value between (0,1), and increases with timestep t, essentially making the final $q(x_T|x_0)$ approximately a simple Gaussian $\mathcal{N}(0, I)$. This forward process is fixed, and the goal of the diffusion model is to learn the reverse process. During training, we simply optimize for predicting the noise $\boldsymbol{\rho}$ from an arbitrary step t in the forward process, forming a loss function [14]

$$L = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\rho}}\left[\|\boldsymbol{\rho} - \boldsymbol{\rho}_\theta\left(x_t, t\right)\|^2\right]. \tag{2}$$

The process can also be conditioned on another variable y in $\boldsymbol{\rho}_\theta(x_t, y, t)$. For example, the diffusion model can be class conditioned [15], with y being a variable representing the class of the sample from a classification dataset. We utilize the class-conditioning approach of [15] in our diffusion models for FL.

2.3 Differential Privacy

Differential privacy (DP) [7–9] is a framework for ensuring that the output of a computation, such as machine learning model training, does not reveal sensitive information about any individual data point in the training dataset. A computation is said to be differentially private if the probability of obtaining a particular output is roughly the same whether a particular individual's data sample is included in the computation or not. Formally [9],

$$Pr[A(D) \in S] \leq e^{\epsilon} \cdot Pr[A(D') \in S] + \delta, \tag{3}$$

where A is a randomized algorithm, D and D' are a pair of datasets that differ in at most one record, and S is any subset of the output space of A. (ϵ, δ) control the level of privacy protection provided by the algorithm, essentially determining the maximum allowable amount of information that can be harnessed from the data. Larger values of (ϵ, δ) correspond to weaker privacy guarantees, while smaller values of (ϵ, δ) correspond to stronger guarantees.

To train deep learning models with such guarantees, differentially private stochastic gradient descent (DPSGD) is typically employed [1]. In DPSGD, two main mechanisms are used to protect the privacy of individual data points: per-sample gradient clipping and the addition of random noise to the clipped gradients. Per-sample gradient clipping involves setting a maximum threshold on the norm of the gradient computed for each data point, so that if the norm of a gradient exceeds the threshold, it is rescaled. This step is necessary to limit the sensitivity of the loss function, which measures how much the loss function changes when a single data point is removed from the training dataset. After the gradients have been clipped, random noise is added to them before they are used to update the model parameters. The amount of noise added is calibrated based on privacy budget parameters (ϵ, δ).

3 Diffusion Models for Federated Learning

Before delving into the underlying motivation for our research questions ***RQ1*** and ***RQ2***, it is essential to provide a brief exposition of the one-shot FL process when integrating generative models. The core premise of this approach departs from the traditional method of client-side discriminative model training. Instead, it advocates for the training of generative models on the client devices. These client-side generative models are aggregated and used offline on the server side to synthesize data, which, in turn, facilitates the training of a global discriminative model. Within the scope of our study, we undertake an investigation into the viability of leveraging diffusion models in this paradigm.

Why diffusion models? In [32], a generative learning trilemma is shown with model types, trading off sample quality, diversity, and fast sampling. CVAEs (as employed in [13]) are typically identified to excel in diversity and fast sampling, but lacking in sample quality. However, for one-shot federated learning, fast sampling is not a concern, as the sampling can be done offline at the server (Figure 1). Therefore, ***high sample quality and diversity are more valuable properties in one-shot FL***, as these will positively impact the performance of the trained global model with the synthetic data. In this trilemma, ***diffusion models excel in sample quality and diversity*** [32], but are not as quick to sample. This motivated us to investigate the potential of DMs in this setting, as the inherit strengths of DMs align with the needs of one-shot FL.

Furthermore, while CVAEs and diffusion models share a common origin in terms of their objective, they differ in their approach to achieving this objective. The optimization task of the diffusion model is simplified to learning a Markov process to reverse a fixed forward process. The training is structured such that the model only needs to learn how to denoise a small step in the generation process, breaking down the problem. In contrast, CVAEs must simultaneously learn both the forward process to encode the image to a latent space, and the decoding process from that latent vector. We reason that the simplified objective of DMs helps achieve superior performance when dealing with complex data within the challenging FL environment (data heterogeneity, class imbalance, and limited sample sizes). Moreover, in the FL setting, privacy is of critical importance. To ensure privacy, training is done with DP, which introduces noise to the training process and increases the difficulty of optimization. In these settings, the simpler training paradigm of diffusion models is potentially advantageous.

Overview. To provide a contextual foundation for our research inquiries, we start by laying out the settings of our study and approach in Section 3.1. With this groundwork, we investigate ***RQ1*** in Section 4, where we delve into the effectiveness of diffusion models in one-shot FL with our FedDiff approach. In Section 5, we address ***RQ2*** through a systematic exploration of one-shot FL methods within provable privacy budgets. Specifically, we evaluate FedDiff and other SOTA approaches under DP constraints, as well as investigate the viability of DP in mitigating memorization. In Section 5.3, we also introduce our Fourier Magnitude Filtering approach, aimed at enhancing the efficacy of generated data for global model training by selectively eliminating low-quality samples.

3.1 FedDiff and Experimental Setup

The basis of our approach, FedDiff, is illustrated in Figure 1. We begin by training class-conditioned diffusion models using the local data $\mathbf{x}$ on the clients. After training, the server collects these local models, denoted as D_0, D_1, ..., D_c, which are then used to generate data $\mathbf{z}_0$, $\mathbf{z}_1$, ..., $\mathbf{z}_c$. The label distributions from the clients are used to condition the generative models during generation, as in [13]. The combination of these synthesized samples forms our global training dataset, $\mathbf{G}$. Subsequently, the global model is trained on the synthetic dataset $\mathbf{G}$ and evaluated in our experiments.

Comparison Methods. We compare with key baselines and the most recent state-of-the-art one-shot FL methods throughout our investigation.

FedAvg [24] is a standard baseline, which simply trains discriminative classifiers at the clients and averages their parameters, typically weighted by the number of samples at each client, to form a single server model.

DENSE [35] is a one-shot FL approach that first trains the discriminative classifiers on the clients to convergence. Once the client models are collected, it performs two stages of training in an interactive manner, switching between training a GAN-based network for generating synthetic data and using the synthetic data to distill the ensemble of client models to a single server model.

OneShot-Ens. We also include an idealized variant of DENSE, where rather than attempt to distill the ensemble of client models to a single server model, we simply employ the ensemble as the final model, as shown in [10] and similarly compared to in [13]. We term this approach OneShot-Ens throughout the paper.

FedCVAE [13]. This recently proposed method employs conditional variational autoencoders (CVAEs) for one-shot federated learning. Their approach has two variants, FedCVAE-KD and FedCVAE-Ens, which differ in how they operate at the server level. FedCVAE-KD distills all generative models from the clients to a single CVAE, and then generates data for training the global model. On the other hand, FedCVAE-Ens employs each client model to generate data, contributing to the final dataset for training the server model. The latter variant always shows significantly better performance than the other in their paper; therefore, we compare with this FedCVAE-Ens variant and refer to it as FedCVAE in the rest of the paper.

Datasets. We employ three datasets, FashionMNIST [31], PathMNIST [33], and CIFAR-10 [18], which provide a range of domains and complexities. More details on the datasets are provided in the **Supplementary Material**. For our experiments, we divide the training set among C clients with a Dirichlet distribution $Dir(\alpha)$, as commonly done in FL literature [4, 12, 13, 25]. This partitioning approach creates imbalanced subsets, where some clients may not have any samples for certain classes. As a result, a significant number of clients will only encounter a small subset (or potentially only one) of the available class instances. We visualize data distributions with $Dir(0.1)$ and $Dir(0.001)$ across 10 clients in the **Supplementary Material**.

Federated Learning Settings. We reproduce DENSE and FedCVAE for our settings with their respective official code repositories. For all experiments, we perform 3 independent runs with different seeds and report the mean and standard deviation. For all approaches, we train client models for 200 local epochs, as in [35]. For DENSE, FedCVAE, and FedDiff, we train the final global model for 50 epochs. For the generator of FedCVAE, we employ their CVAE variant with residual blocks, which has approximately 5.9M parameters. For our diffusion model, we employ a basic U-Net structure with residual blocks [14, 27] and class-conditioning, with similar parameters to FedCVAE ($\sim$5.8M). We employ a ResNet16 architecture for the discriminative models with approximately 6.4M parameters. For experiments with differential privacy, we employ the Opa-

cus [34] library in PyTorch [26] to track privacy budgets. Further training details and additional experiments can be found in **Supplementary Material**.

4 *RQ1*: FedDiff for One-Shot FL

We investigate ***RQ1*** by exploring the efficacy of our FedDiff approach and other SOTA one-shot FL methods across important FL scenarios, including different data heterogeneity levels, number of clients, and resource requirements.

4.1 Data Heterogeneity

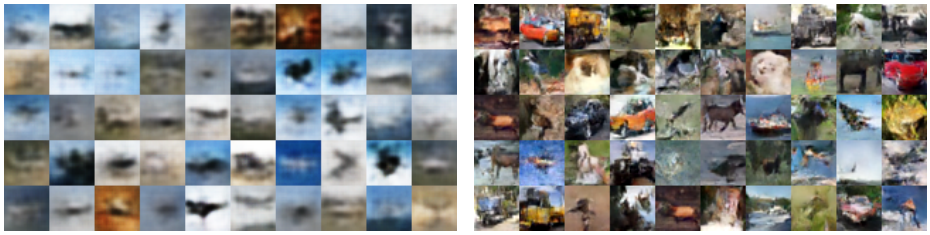
Data heterogeneity is a critical challenge in FL, particularly with one-shot settings. Even in the standard FL scenario of multiple communication rounds, client models often fit to very different distributions, and effectively reconciling their learnings is daunting. This is exacerbated in the one-shot setting, as we no longer have the luxury of getting many iterations to progressively steer the learning process towards an ideal encompassing representation.

Table 1: Data heterogeneity results with various $Dir(\alpha)$ partitions. Smaller alpha values indicate higher levels of heterogeneity. Typical approaches leveraging discriminative models rapidly degrade in performance as heterogeneity increases. However, generative approaches are more robust to such conditions. **Our FedDiff shows superior performance to all, particularly in the most challenging scenarios (CIFAR-10, high heterogeneity)**.

	Method	$\alpha = 0.1$	$\alpha = 0.01$	$\alpha = 0.001$
FashionMNIST	FedAvg	57.11±3.64	29.50±10.6	25.89±4.78
	DENSE	65.20±3.55	28.92±17.3	27.68±4.08
	OneShot-Ens	67.35±1.19	33.79±17.9	32.01±3.35
	FedCVAE	78.08±2.69	78.81±3.25	81.53±0.23
	FedDiff	**87.21±0.74**	**86.81±0.54**	**86.59±0.69**
PathMNIST	FedAvg	28.10±4.60	22.05±8.20	21.92±4.95
	DENSE	50.97±3.19	29.26±10.7	27.69±4.52
	OneShot-Ens	34.62±3.61	34.94±9.32	34.49±5.30
	FedCVAE	41.60±0.82	44.81±1.41	47.35±3.21
	FedDiff	**74.58±1.02**	**70.61±1.37**	**69.43±1.30**
CIFAR-10	FedAvg	19.64±2.39	19.01±3.76	18.16±5.49
	DENSE	36.04±7.75	21.40±2.73	17.91±3.18
	OneShot-Ens	39.38±7.53	23.38±3.62	20.15±9.11
	FedCVAE	34.40±1.04	36.06±3.27	36.92±1.38
	FedDiff	**57.69±2.07**	**56.57±2.42**	**55.75±1.55**

In Table 1, we analyze the performance of all methods under moderate ($Dir(0.1)$) to extreme ($Dir(0.001)$ heterogeneity. Interestingly, FedDiff outperforms all other methods by a significant margin, from ∼5% to up to ∼20% in different scenarios. In the case of CIFAR-10, which is the most complex of the datasets, we find that FedDiff provides the most improvement. As discussed in our initial motivations (Section 3), we reason that the focus on sample quality and diversity that is provided by the DM objective enables much improved performance. Intuitively, this becomes increasingly evident in more complex settings. To verify this observation, we conduct a comparative analysis of generated samples produced by FedCVAE and our FedDiff approach, illustrated in Figure 2. The discernible disparity is evident, with the samples generated by our method exhibiting significantly enhanced sharpness and overall quality.

4.2 Number of Clients



(a) FedCVAE **(b)** FedDiff (ours)

Fig. 2: Random sets of generated samples from FedCVAE and our FedDiff approach. By leveraging the intrinsic properties of diffusion models (DMs), which are well-aligned with the requirements of one-shot FL, we achieve substantial benefits in sample quality and subsequent global model performance.

Deepening our investigation, we also study the effect of the number of clients C in Table 2. Note that, as we employ the same total number of samples in all experiments, the number of samples *per client* will increase with smaller C, and decrease with larger C. This allows us to observe the effect of increasing the distributed nature of the data across the client network.

Table 2: Results with varying number of clients C with $Dir(0.01)$. As a fixed-size dataset is used in all experiments, increasing the number of clients also decreases the number of samples per client. We find that the SOTA discriminative approaches quickly degrade as the data is distributed across more clients. On the contrary, **our FedDiff maintains strong performance in all settings**.

	Method	$C = 5$	$C = 10$	$C = 20$
FashionMNIST	FedAvg	43.73±2.37	29.50±10.6	28.38±3.17
	DENSE	48.24±6.25	28.92±17.3	20.72±9.82
	OneShot-Ens	49.53±6.08	33.79±17.9	31.36±8.01
	FedCVAE	78.45±2.44	78.81±3.25	78.33±2.45
	FedDiff	**86.89±0.34**	**86.81±0.54**	**87.24±0.57**
PathAMNIST	FedAvg	29.18±3.54	22.05±8.20	19.83±3.16
	DENSE	33.39±6.14	29.26±10.7	20.79±5.77
	OneShot-Ens	36.95±5.30	34.94±9.32	24.59±6.19
	FedCVAE	46.16±1.17	44.81±1.41	41.25±1.68
	FedDiff	**72.74±0.63**	**70.61±1.37**	**69.11±0.99**
CIFAR-10	FedAvg	29.14±5.15	19.24±3.77	15.79±2.64
	DENSE	30.48±2.30	21.40±2.73	12.60±2.33
	OneShot-Ens	36.17±3.21	23.38±3.62	13.23±2.96
	FedCVAE	32.34±2.59	36.06±3.27	37.63±1.87
	FedDiff	**57.68±1.86**	**56.57±2.42**	**58.45±0.73**

One question arising from the adoption of generative models in FL settings pertains to their ability to maintain satisfactory performance when trained on a limited number of samples. Interestingly, when analyzing the results, we find that FedDiff is capable of handling a much smaller number of client training samples with little to no performance degradation. On the other hand, the discriminative model approaches quickly experience a collapse in performance when expanding to 20 clients. In the heterogeneous environment of federated learning, the local optimization of a discriminative model on a highly-imbalanced and small dataset proves challenging. Rather than being an overwelming burden, such a situation is handled well by FedDiff, as its sole focus is to capture the subsequently smaller distribution. Furthermore, we again find that FedDiff outperforms FedCVAE in all settings, further illustrating the potential for diffusion models in one-shot FL.

4.3 Resource Requirements

To further explore the efficacy of our method, we also examine resource factors, including FLOPs and parameter count, for each method deployed on a single client. Notably, our FedDiff approach consistently delivers superior accuracy with comparable computational resources to other methods. We extend this assessment to a reduced model size (FedDiff$_s$ in Table 3), reaffirming its strong performance relative to alternative methods. This analysis underscores the effectiveness of FedDiff, even when deployed on hardware with modest computational capabilities.

It is pertinent to emphasize that training diffusion models within the FedDiff framework is *no more intricate than conventional methodologies and remains highly viable for FL*. The computational complexity aligns with training a conventional CNN model with a modest number of parameters, and we employ the same number of local epochs as previous work with CNNs [35]. Importantly, the training process entails selecting random steps in the diffusion process at any given training iteration, eliminating the necessity for sequential steps during training. During inference, the generation process involves sequential denoising steps; however, this poses no issue for the clients, as generation occurs at the server in FedDiff. Therefore, FedDiff is an effective and practical approach for providing strong performance.

Table 3: Accuracy versus FLOPs and parameter count (Params) for each method on a single client. Our FedDiff approach consistently attains heightened accuracy levels while maintaining very reasonable resource demands on par with other methodologies. We also evaluate our method with a scaled-down model variant (FedDiff$_s$), further confirming its performance relative to alternative approaches. This analysis underscores the realistic feasibility of our FedDiff framework.

	Resources		Accuracy		
Method	MFLOPs ↓	Params ↓	FashionMNIST	PathMNIST	CIFAR-10
FedAvg	479.92	6.44M	29.50±10.6	22.05±8.20	19.24±3.77
DENSE	479.92	6.44M	28.92±17.3	29.26±10.7	21.40±2.73
OneShot-Ens	479.92	6.44M	33.79±17.9	34.94±9.32	23.38±3.62
FedCVAE	79.00	5.97M	78.81±3.25	44.81±1.41	36.06±3.27
FedDiff	301.14	5.81M	**86.81±0.54**	**70.61±1.37**	**56.57±2.42**
FedDiff$_s$	**77.43**	**1.46M**	**85.90±0.92**	**70.53±5.61**	**50.08±1.87**

5 *RQ2*: Privacy Considerations

Privacy holds paramount importance in one-shot FL. The trained client model may be repeatedly utilized, or even exchanged in a model market context, and therefore safeguarding the privacy of the model before it leaves the client is imperative. However, other SOTA works have not experimented with DP constraints, nor have they thoroughly explored this aspect, often leaving privacy discussions simply as a possibility for future work [13, 35]. In the subsequent sections, we meticulously investigate privacy from various perspectives and delve into our research question ***RQ2***.

5.1 Differential Privacy

Differential privacy is the widely accepted standard for ensuring privacy of a model, as it offers a provable guarantee of privacy [1, 7–9, 11]. Utilizing (ϵ, δ) dif-

ferential privacy during model training guarantees comprehensive privacy protection, encompassing not only the model's parameters and activations, but also extending to all subsequent downstream operations such as inferences, fine-tuning, and distillation. It is important to note that a model trained under (ϵ, δ) DP **safeguards the privacy of every training sample, regardless of its qualities or uniqueness** [34]. Specifically, we train all approaches under (ϵ, δ) DP at the clients for various privacy levels of $\epsilon = 50$, 25, and 10, with $\delta = 10e^{-5}$, $C = 10$, and $\alpha = 0.01$. Lower ϵ values correspond to a tighter privacy budget, and the stated budget is for the entire training of each local model. We employ the Opacus [34] library for implementing DP. Further DP training details are provided in **Supplementary Material**, along with additional ϵ experiment. We present the results for all approaches in Table 4.

As expected, all methods experience a drop in performance when trained under DP settings. Nonetheless, FedDiff still stands out, outperforming all other methods by a significant margin. Particularly for FashionMNIST, FedDiff experiences comparatively less accuracy drop under DP than FedCVAE. As articulated in our initial motivations outlined in Section 3, DP training introduces noise into the training process, exacerbating the complexity of optimization. In such scenarios, the simplicity of the training paradigm employed by diffusion models becomes notably advantageous. Overall, we show that FedDiff is a strong approach even when DP is employed.

Table 4: Differential privacy (DP) results under various ϵ budgets. We set $C = 10$ and $\alpha = 0.01$ as the default setting. **Even under DP constraints, FedDiff is a particularly viable approach, outperforming all other SOTA one-shot FL methods**.

	Method	$\epsilon = 50$	$\epsilon = 25$	$\epsilon = 10$
FashionMNIST	FedAvg	21.04±12.1	20.82±12.3	20.39±12.6
	DENSE	26.34±9.03	26.29±9.81	24.29±15.6
	OneShot-Ens	31.27±10.9	31.32±10.1	29.99±16.7
	FedCVAE	44.40±1.70	43.89±2.53	41.65±3.19
	FedDiff	**75.92±1.86**	**75.08±2.13**	**73.43±1.50**
PathMNIST	FedAvg	16.98±8.93	15.30±6.44	14.85±4.19
	DENSE	20.56±6.59	19.19±3.76	18.41±1.86
	OneShot-Ens	24.59±7.63	23.38±2.60	22.23±2.02
	FedCVAE	24.06±1.57	22.15±2.68	20.51±1.29
	FedDiff	**54.98±2.04**	**51.51±1.85**	**47.85±3.68**
CIFAR-10	FedAvg	16.35±1.52	15.39±1.87	15.07±2.12
	DENSE	16.97±2.35	15.68±2.27	14.98±1.25
	OneShot-Ens	17.73±2.71	17.34±2.35	15.72±1.34
	FedCVAE	16.29±1.55	16.08±2.19	15.86±2.83
	FedDiff	**32.93±1.93**	**31.76±2.68**	**27.78±1.66**

5.2 Addressing Memorization

In a recent study, [5] explored diffusion models and identified their ability to memorize samples under certain conditions. They acknowledge differential privacy as the gold standard defense strategy, but did not provide completed experiments to this end. Therefore, we evaluate the effectiveness of DP to this end, assessing memorization within our DP-trained models to investigate whether inadvertent reproduction of the training data can be eliminated.

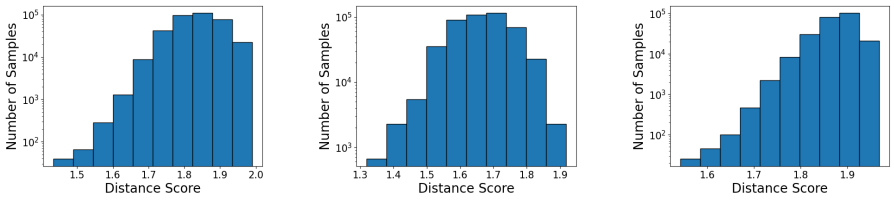
To conduct this study, we adopt the evaluation methodology established by [5] to scrutinize the occurrence of memorization. Specifically, from each DP-trained diffusion model, we generate a vast number of samples (five times the size of the training set). Subsequently, for each generated image, we assess potential memorization compared to the original training samples using the adaptive

distance metric introduced by [5],

$$\ell\left(\hat{x}, x; S_{\hat{x}}\right) = \frac{\ell_2(\hat{x}, x)}{\alpha \cdot \mathbb{E}_{y \in S\hat{x}}\left[\ell_2(\hat{x}, y)\right]}. \tag{4}$$

Here, $S_{\hat{x}}$ denotes the set comprising the n nearest elements from the training dataset to the example $\hat{x}$. The resulting distance metric yields a small value if the extracted image x exhibits significantly closer proximity to the training image $\hat{x}$ compared to the n closest neighbors of $\hat{x}$ within the training set. The idea is to find generated images that are unusually close to an original training image as indication of memorization. We set $\alpha = 0.5$ and $n = 50$ as in [5].

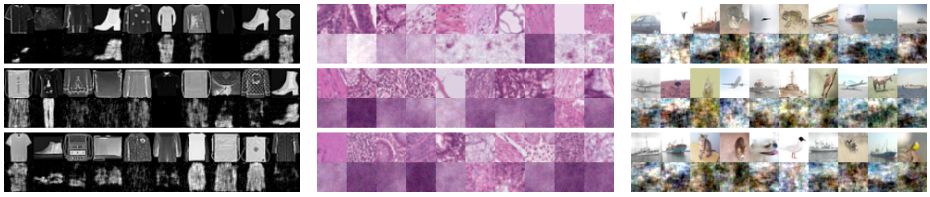
[5] did not define the specific threshold for Equation 4 for marking when a sample is considered memorized. Therefore, we consider the intuitive threshold to be less than 1, as this would indicate that the distance from the extracted image to the training image is less than half of the average distance to the closest n neighbors. Upon conducting this assessment, we do not find any instances of memorized samples for all datasets under such definition, even at an elevated privacy parameter of $\epsilon = 50$, with the closest distance values being $\sim$1.3. We show the histogram of scores for all samples on each dataset in Figure 3.



(a) FashionMNIST **(b)** PathMNIST **(c)** CIFAR-10

Fig. 3: Histogram of distance scores for all generated samples at $\epsilon = 50$ to corresponding closest training image by Eq. 4 on each dataset. Note that the y-axis in in *log scale*, as there are very few samples with lower scores.

Because the threshold definition for memorization could vary, we also qualitatively show the samples with the lowest distances for all datasets at $\epsilon = 50$ in Figure 4. Notably, the training versus the generated samples have discernible differences, in contrast to the nearly identical samples uncovered in [5] when training large diffusion models without DP. Also, given the nature of FL, the choice of diffusion model size will typically be small (for example, ours is $\sim$5.8M parameters), and therefore will be less likely to memorize compared to the larger DMs evaluated in [5]. As DP algorithms improve, we anticipate that even better final accuracy can be achieved while maintaining guaranteed privacy in the future with FedDiff.



(a) FashionMNIST **(b)** PathMNIST **(c)** CIFAR-10

Fig. 4: Qualitative comparison of original training samples and generated samples at $\epsilon = 50$. We show the closest 30 samples via the similarity metric in Equation 4. In each stacked row, the original samples are on top, with the corresponding nearest generated image immediately below. Even under the loosest privacy guarantee of $\epsilon = 50$, we do not see blatant memorization.

5.3 Fourier Magnitude Filtering

While FedDiff performs comparatively well against other SOTA one-shot FL methods under DP constraints, we further investigate a simple approach to improve our method, particularly for complex data most affected by DP. As shown in Figure 4, we note that the generated samples under DP can lack details, exhibiting reduced structure. Therefore, it may be advantageous to sort out and remove such poor quality samples from the final synthetic dataset prior to conducting the training of the global model.

In order to understand the impact of prioritizing data quality on performance, we conducted an initial experiment. For the CIFAR-10 dataset, we leverage a centralized pretrained classifier as an oracle to discern high and low quality samples. Specifically, we selectively retain samples for which the oracle accurately classifies and discarded those it misclassifies. This provides a way to filter out samples that are likely irrelevant or misleading for training a model. We then train the global model exclusively on the curated dataset of accurate samples and evaluate. This investigation yields a discernible improvement ranging from approximately 2% to 4% in final global model accuracy compared to training with all generated data, verifying an importance for data quality. Therefore, a critical question arises from this observation: how can we conduct sample filtration in the absence of an oracle?

To do so, we look to the Fourier domain for a potential source of information. As inspiration, we note that the use of the magnitude of local client images has been utilized in FL to assist in domain generalization across clients by providing low-level “style" information without the high-level semantics encoded in the phase [23]. In our case, we propose to leverage the Fourier magnitude information as a potential referenceable indicator to guide the sample filtering process. Furthermore, we are able to do so under very tight DP guarantees.

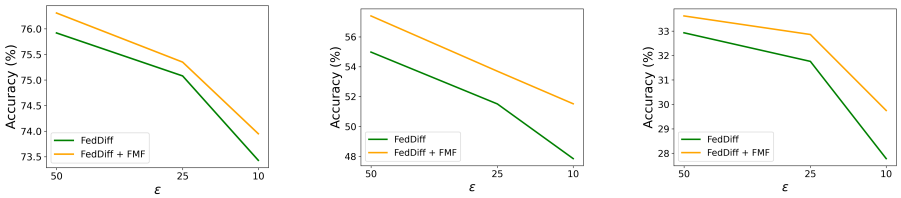
Specifically, on the client, we take the Fourier transform of the local samples and extract the magnitude information. For each client c, we gather the average

sample magnitude with

$$\bar{M}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} |\psi(\mathbf{x}^i)|, \tag{5}$$

where ψ is the 2D Fourier transform operation, $\mathbf{x}^i$ is a sample, and n_c is the total number of samples in client c. $\bar{M}$ is bundled with the model and transmitted by the client to the relevant global party.

As in our standard global training procedure, samples are generated with the client-trained diffusion models to form a synthetic set. Prior to conducting global training, we calculate a sample score s for the generated data z from each diffusion model from the clients, $s_{\mathbf{z}_c^i} = \| |\psi(\mathbf{z}_c^i)| - \bar{M}_c \|_2$. We can then leverage this information to guide the removal of irrelevant samples, forming the final training set $\mathbf{G}$ by removing γ percent of the generated data with the highest s (larger magnitude difference). To continually ensure privacy guarantees, we apply DP in the FMF calculation. We do so by employing the DP bounded mean [19] from PyDP[1] to calculate the average magnitude $\bar{M}_c$ at each client. This allows us to precisely manage any degree of privacy leakage for $\bar{M}_c$ and include it in the overall privacy budget.



(a) FashionMNIST **(b)** PathMNSIT **(c)** CIFAR-10

Fig. 5: Results with our Fourier Magnitude Filtering under DP. FedDiff is in **green** and FedDiff+FMF in **orange**. Our FMF approach provides a simple way to boost accuracy, especially in more challenging scenarios such as lower ϵ budgets and more complex datasets. We plot the mean across three runs with different seeds for each setting. Additional γ ablations are provided in the **Supplementary Material**.

In Figure 5, we show the results of applying our FMF approach with FedDiff for the same overall DP budgets as Table 4. FMF is particularly effective in the most difficult scenarios, helping to mitigate the performance drop in harsh FL environments. For example, FMF provides over 3.5% and 2% improvements with PathMNIST and CIFAR-10 in the challenging $\epsilon = 10$ setting. Overall, FMF is a simple way to boost performance in one-shot FL under DP.

6 Conclusion

In conclusion, our work addresses two valuable research questions in one-shot FL. Firstly, we investigate the potential of diffusion models for one-shot FL, and

[1] https://github.com/OpenMined/PyDP

present the pioneering effort in this direction. In our investigation, we unveil the unique advantages that DMs offer, showcasing their potential to enhance the overall performance and tackle heterogeneity across diverse settings with our proposed approach, FedDiff. Secondly, we study privacy in SOTA one-shot FL and contribute a thorough investigation under provable privacy budgets, as well as address memorization concerns. Furthermore, to enhance performance under harsh DP conditions, we propose a novel and pragmatic solution, Fourier Magnitude Filtering, to boost the efficacy of generated data for global model training by eliminating low-quality samples. More discussions, including limitations and broader impact, are included in the **Supplementary Material**. We hope our work will inspire the community and foster further research in this direction to improve one-shot FL with generative models.

References

1. 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. CCS'16, ACM (Oct 2016). https://doi.org/10.1145/2976749.2978318, http://dx.doi.org/10.1145/2976749.2978318
2. Abay, A., Zhou, Y., Baracaldo, N., Rajamoni, S., Chuba, E., Ludwig, H.: Mitigating bias in federated learning. CoRR **abs/2012.02447** (2020), https://arxiv.org/abs/2012.02447
3. Acar, D.A.E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., Saligrama, V.: Federated learning based on dynamic regularization. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=B7v4QMR6Z9w
4. Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N., Saligrama, V.: Federated learning based on dynamic regularization. CoRR **abs/2111.04263** (2021), https://arxiv.org/abs/2111.04263
5. Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., Wallace, E.: Extracting training data from diffusion models. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 5253–5270 (2023)
6. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. CoRR **abs/2105.05233** (2021), https://arxiv.org/abs/2105.05233
7. Dwork, C.: A firm foundation for private data analysis. Communications of the ACM **54**(1), 86–95 (2011)
8. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3. pp. 265–284. Springer (2006)
9. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
10. Guha, N., Talwalkar, A., Smith, V.: One-shot federated learning. arXiv preprint arXiv:1902.11175 (2019)
11. Ha, T., Dang, T.K., Dang, T.T., Truong, T.A., Nguyen, M.T.: Differential privacy in deep learning: An overview. In: 2019 International Conference on Advanced

Computing and Applications (ACOMP). pp. 97–102 (2019). https://doi.org/10.1109/ACOMP.2019.00022
12. He, C., Li, S., So, J., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Shen, L., Zhao, P., Kang, Y., Liu, Y., Raskar, R., Yang, Q., Annavaram, M., Avestimehr, S.: Fedml: A research library and benchmark for federated machine learning. CoRR **abs/2007.13518** (2020), https://arxiv.org/abs/2007.13518
13. Heinbaugh, C.E., Luz-Ricca, E., Shao, H.: Data-free one-shot federated learning under very high statistical heterogeneity. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=_hb4vM3jspB
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. CoRR **abs/2006.11239** (2020), https://arxiv.org/abs/2006.11239
15. Ho, J., Salimans, T.: Classifier-free diffusion guidance (2022)
16. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K.A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., Rouayheb, S.E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S.U., Sun, Z., Suresh, A.T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F.X., Yu, H., Zhao, S.: Advances and open problems in federated learning. CoRR **abs/1912.04977** (2019), http://arxiv.org/abs/1912.04977
17. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S.J., Stich, S.U., Suresh, A.T.: SCAFFOLD: stochastic controlled averaging for on-device federated learning. CoRR **abs/1910.06378** (2019), http://arxiv.org/abs/1910.06378
18. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
19. Li, N., Lyu, M., Su, D., Yang, W.: Differential privacy: From theory to practice. Synthesis Lectures on Information Security, Privacy, & Trust **8**(4), 1–138 (2016)
20. Li, Q., He, B., Song, D.: Model-agnostic round-optimal federated learning via knowledge transfer. CoRR **abs/2010.01017** (2020), https://arxiv.org/abs/2010.01017
21. Li, Q., He, B., Song, D.: Model-contrastive federated learning. CoRR **abs/2103.16257** (2021), https://arxiv.org/abs/2103.16257
22. Liu, P., Xu, X., Wang, W.: Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. Cybersecurity **5**(1), 1–19 (2022)
23. 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. The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
24. McMahan, H.B., Moore, E., Ramage, D., y Arcas, B.A.: Federated learning of deep networks using model averaging. CoRR **abs/1602.05629** (2016), http://arxiv.org/abs/1602.05629
25. Mendieta, M., Yang, T., Wang, P., Lee, M., Ding, Z., Chen, C.: Local learning matters: Rethinking data heterogeneity in federated learning. CoRR **abs/2111.14213** (2021), https://arxiv.org/abs/2111.14213
26. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)

27. Ronneberger, O., Fischer, P., Brox, T.: 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. pp. 234–241. Springer (2015)
28. Sahu, A.K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., Smith, V.: On the convergence of federated optimization in heterogeneous networks. CoRR **abs/1812.06127** (2018), http://arxiv.org/abs/1812.06127
29. Salehkaleybar, S., Sharifnassab, A., Golestani, S.J.: One-shot federated learning: theoretical limits and algorithms to achieve them. The Journal of Machine Learning Research **22**(1), 8485–8531 (2021)
30. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015), https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf
31. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. CoRR **abs/1708.07747** (2017), http://arxiv.org/abs/1708.07747
32. Xiao, Z., Kreis, K., Vahdat, A.: Tackling the generative learning trilemma with denoising diffusion gans. CoRR **abs/2112.07804** (2021), https://arxiv.org/abs/2112.07804
33. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Scientific Data **10**(1), 41 (2023)
34. Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., Mironov, I.: Opacus: User-friendly differential privacy library in PyTorch. arXiv preprint arXiv:2109.12298 (2021)
35. Zhang, J., Chen, C., Li, B., Lyu, L., Wu, S., Ding, S., Shen, C., Wu, C.: Dense: Data-free one-shot federated learning. Advances in Neural Information Processing Systems **35**, 21414–21428 (2022)
36. Zhou, Y., Pu, G., Ma, X., Li, X., Wu, D.: Distilled one-shot federated learning. arXiv preprint arXiv:2009.07999 (2020)

7 Supplementary Material

7.1 Additional Training Details

The FashionMNIST dataset is an alternative to the original MNIST dataset, providing a more challenging task by replacing the handwritten digits with grayscale images of various fashion items. The dataset consists of 60,000 training images and 10,000 test images. The PathMNIST dataset is a medical dataset of colon pathology images in RGB, with a training set of 89,996 images and a test set containing 7,180 images with 9 classes. The CIFAR-10 dataset consists of 60,000 color images equally distributed into ten different classes. The dataset is composed of a training set containing 50,000 images and a test set comprising of 10,000 images. CIFAR-10 is natively sized at 32×32 pixels. We upsample FashionMNIST and PathMNIST from 28×28 to 32×32. A visualization of the dataset partitioning across clients is shown in Figure 6.

We train with a batch size of 128 for all methods and use the AdamW optimizer. For local (and global training were applicable), we searched learning rates from [$3e^{-3}$, $1e^{-3}$, $3e^{-4}$, $1e^{-4}$] for each method using the CIFAR-10 dataset to find the optimal settings. For DP experiments, we set the max gradient norm clipping threshold to 1.0 for all experiments and methods. In accordance with the recommendations of the Opacus [34] library, we employ their Poisson batch sampling to ensure privacy guarantees.

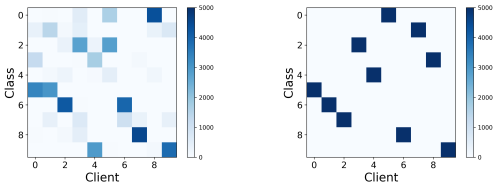
As mentioned in Section 3.1 of the main paper, our diffusion model is a basic U-Net structure with residual blocks [14,27] and class-conditioning. For sampling at the server, we perform 1000 iterations as in [14] to generate each batch. The total number of generated samples is set equal to the size of the original dataset. Code will be made available upon acceptance.

7.2 Additional ϵ Experiment

To demonstrate the feasibility of FedDiff under more stringent budget constraints, we conduct an experiment with an even tighter privacy budget of $\epsilon = 1$ in Table 5. Despite facing such stringent privacy constraints, FedDiff maintains a higher level of performance at $\epsilon = 1$ than all other methods in Table 4 of the main paper at $\epsilon = 50$.

7.3 FMF γ Ablation

In Figure 7, we present the outcomes obtained using FedDiff+FMF under $\epsilon = 10$ across a range of γ values, encompassing data filtering percentages spanning from 1% to 12%. Our findings indicate that, in general, data filtering within the 1% to 10% range yields favorable results and leads to performance enhancements, with around 5% being a great default. Interestingly, the the degree of improvement provided by FMF becomes more pronounced and consistent as the dataset becomes more challenging. This phenomenon aligns with the anticipated trends, as more intricate datasets inherently pose a greater challenge, making it less likely

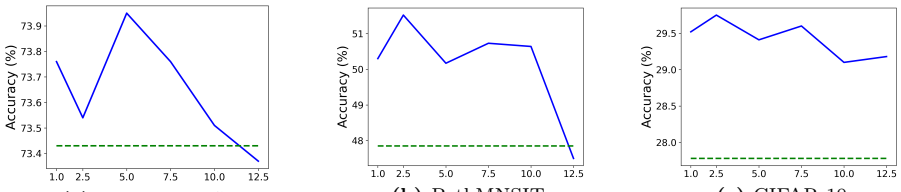


(a) $\alpha = 0.1$ **(b)** $\alpha = 0.001$

Fig. 6: $Dir(\alpha)$ data partitioning for 10 clients on CIFAR-10. We show moderate ($\alpha = 0.1$) to severe ($\alpha = 0.01$) data heterogeneity levels. Data heterogeneity poses a significant challenge for many one-shot FL methods, as reconciling various models trained on widely different distributions is non-trivial. Our FedDiff approach rather trains diffusion models on the simple client distributions, which can then generate useful synthetic data for training global models.

Table 5: Differential privacy results under $\epsilon = 1$. Even in this case, FedDiff outperforms all other methods with the much larger budget of $\epsilon = 50$ in Table 4 of the main paper.

Privacy $\epsilon = 1$	**FedDiff**
FashionMNIST	**65.53±0.70**
PathMNIST	**44.38±3.35**
CIFAR-10	**21.48±1.53**



(a) FashionMNIST **(b)** PathMNSIT **(c)** CIFAR-10

Fig. 7: Ablation study of γ in FMF under the $\epsilon = 10$ setting. The accuracy of FedDiff is in **green** and FedDiff+FMF for various γ in **blue**. Generally, data filtering within the range of 1% to 10% produces positive outcomes, resulting in improved performance, with approximately 5% serving as an effective default choice. We plot the mean across three runs with different seeds for each setting.

for the generators to consistently produce high-quality samples. Consequently, the need for data filtering becomes more pronounced in such scenarios to enhance sample quality. This trend is also favorable since it addresses the specific need for improvement, especially in cases where performance is suboptimal and the challenges are more pronounced.

7.4 Discussions, Limitations and Broader Impact

Model Heterogeneity. In real FL systems, model heterogeneity may often occur [13, 35]. For instance, some clients may have architecture variations in their models or have smaller or larger models depending on their computing capabilities. Therefore, clients may have different architectures of similar generation capability, or even differing capabilities depending on the requirements of each

client. Our approach allows for flexibility to accommodate such system diversity across clients. In FedDiff, we generate data from the client models and employ that synthetic data for global training, and therefore can leverage varying models without the worry of reconciling the weights themselves.

Limitations and Broader Impact. One downside of our method is that the generated data, particularly under DP constraints, still lacks in quality and effectiveness for global model training versus using true data. For instance, with DP on CIFAR-10 as shown in Figure 4 in the main paper, the data loses a substantial amount of structure. An interesting direction for future work would be to study how to further improve the quality of the generated data and its usefulness for global model training while maintaining privacy.

Looking at the broader impact of our work, FL depends on the diversity of data contributed by different participants. If biases exist in the local datasets, they can be propagated and amplified during the model training process. This could lead to unintended algorithmic biases and discrimination in the resulting models. Ensuring diversity and fairness in the data used for FL is an important research direction to mitigate this risk and promote equitable outcomes [2], particularly in the highly data heterogeneous environments explored in this work. Furthermore, as we have discussed throughout our paper, the privacy of client data is important in FL. To mitigate risks in this regard, we take many precautions to preserve privacy of the clients participated in the FL process though the use of DP, and operating within the one-shot setting to reduce the chance of eavesdropping.