One-shot Federated Learning via Synthetic Distiller-Distillate Communication

Junyuan Zhang[1,2] **Songhua Liu**[1] **Xinchao Wang**[1*]
National University of Singapore[1] Beihang University[2]
junyuanpk@gmail.com, songhua.liu@u.nus.edu, xinchao@nus.edu.sg

Abstract

One-shot Federated learning (FL) is a powerful technology facilitating collaborative training of machine learning models in a single round of communication. While its superiority lies in communication efficiency and privacy preservation compared to iterative FL, one-shot FL often compromises model performance. Prior research has primarily focused on employing data-free knowledge distillation to optimize data generators and ensemble models for better aggregating local knowledge into the server model. Prior research has primarily focused on employing data-free knowledge distillation to optimize data generators and ensemble models for better aggregating local knowledge into the server model. However, these methods typically struggle with data heterogeneity, where inconsistent local data distributions can cause teachers to provide misleading knowledge. Additionally, they may encounter scalability issues with complex datasets due to inherent two-step information loss: first, during local training (from data to model), and second, when transferring knowledge to the server model (from model to inversed data). In this paper, we propose FedSD2C, a novel and practical one-shot FL framework designed to address these challenges. FedSD2C introduces a distiller to synthesize informative distillates directly from local data to reduce information loss and proposes sharing synthetic distillates instead of inconsistent local models to tackle data heterogeneity. Our empirical results demonstrate that FedSD2C consistently outperforms other one-shot FL methods with more complex and real datasets, achieving up to 2.6 × the performance of the best baseline. Code: https://github.com/Carkham/FedSD2C

1 Introduction

Federated learning (FL) has emerged as a cutting-edge technology that enables training a global model across multiple clients without sharing their raw data [1]. Original FL requires multiple communication rounds for exchanging information between clients and servers. While this paradigm yields a better global model by frequent communication, such high communication costs along with the risk of connection drop errors make it impractical and intolerable in real-world FL applications [2, 3, 4]. Moreover, frequent communication poses security risks such as man-in-the-middle attacks [5] and privacy concerns [6].

To address these issues, one-shot FL [7] has been proposed, requiring only a single communication round, significantly reducing communication costs and concurrently diminishing vulnerability to malicious interception. Due to one communication round property, One-shot FL can also be easily scaled up to large-scale client scenarios, especially for cross-device settings [3]. Despite its sufficient benefits, the limitation of a single communication round makes one-shot FL fall short in accuracy compared to conventional multiple-round FL.

[*] Corresponding Author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-3259

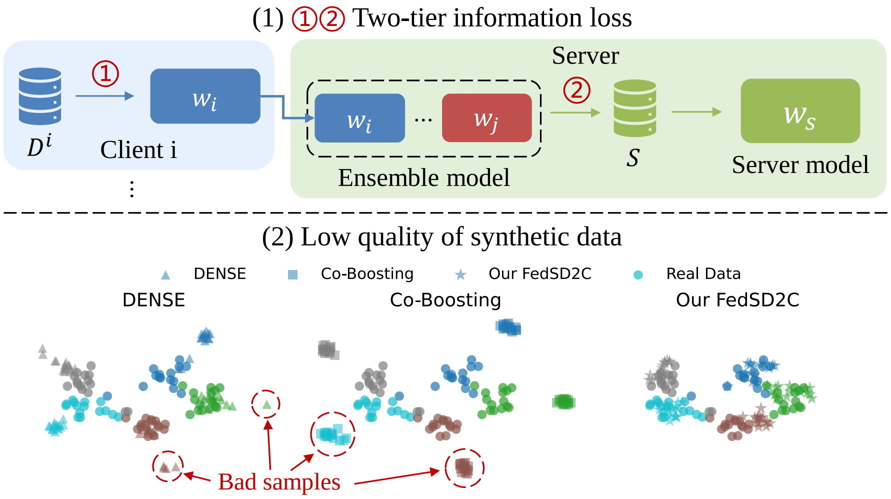


Figure 1: Illustration of issues in one-shot FL based on DFKD: (1) Information loss occurs during the transfer from local data to the model and from the model back to the inversed data. (2) t-SNE plots of feature distributions of data generated by DENSE(left ▲), Co-Boosting(middle ■), and our FedSD2C(right ★). We randomly select five different classes (indicated by different colors) of real and synthetic data from Tiny-ImageNet. Bad samples are data generated by the DFKD-based method that deviates from the distribution of local real data.

To facilitate effective knowledge transfer from client models to the server model within a single round, most previous one-shot FL methods [7, 8, 9, 10] have focused on knowledge distillation. Early approaches [11, 7] use knowledge distillation to transfer knowledge from an ensemble of client models [12] to the server model. While effective, these methods often require additional public datasets, which can be cumbersome or unfeasible in real-world scenarios [13]. Alternatively, data-free knowledge distillation (DFKD) is introduced to avoid the need for public datasets. For example, DENSE [8] employs Generative Adversarial Networks (GANs) [14] as data generators and an ensemble of client models as a discriminator to synthesize diverse data for knowledge transfer to server model in a data-free manner. Co-Boosting [10] extends DENSE by proposing a mutually reinforcing approach to enhance synthetic data and the ensemble model for server training.

Nevertheless, such a method of generating data implies two-tier information loss. First, due to model capacity limitations, client models may struggle to encapsulate all information about local data, affecting the quality of the generated data. Second, the generated data do not fully represent the information within the model, as they are produced from random noise without explicit guidance. Therefore, some classes of synthetic data cannot have similar feature distributions to the original real data, as depicted in Figure 1. Moreover, data heterogeneity [15] in FL can result in inconsistent and misleading predictions from local models [16], which has been shown to hinder knowledge distillation [17, 18]. Consequently, the server model trained on such noisy and information-lossy generated data typically suffers significant performance degradation, particularly on complex datasets [19].

In this paper, we propose FedSD2C (One-shot **Fed**erated Learning via **S**ynthetic **D**istiller-**D**istillate **C**ommunication), a novel and practical one-shot FL framework that introduces a pre-defined distiller for informative, privacy-enhanced, and communication-efficient distillate communication. In specific, FedSD2C first adopts a $\mathcal{V}$-information [20] based Core-Set selection method to distill the local dataset into an informative Core-Set. By capturing the diversity and realism through $\mathcal{V}$-information, the distilled Core-Set fully encapsulates the information of the local data domain for training a robust server model. However, directly transmitting the Core-Set, which may include the original samples, poses potential privacy risks and incurs significant communication costs, especially for high-resolution images. In this regard, FedSD2C employs two techniques to further distill the Core-Set into distillates, thereby enhancing privacy and reducing communication costs: 1) Utilizing Fourier transform perturbation to alter the amplitude components of the Core-Set samples for distillate initialization, enhancing privacy while retaining semantic content; 2) Employing a pre-trained Autoencoder [21] provided by the server as a distiller to distill the perturbed Core-Set into distillates and optimizing its $\mathcal{V}$-information to be as close as possible to original Core-Set, thus minimizing information loss. Finally, clients transmit synthetic distillates to the server instead of inconsistent models for knowledge transfer. Compared to generating noisy knowledge from inconsistent client models with two-tier information loss, end-to-end distillate synthesis minimizes information loss and their aggregation mitigates the impact of data heterogeneity. Through extensive experiments over various real-world datasets , we show that our proposed method significantly surpasses the generated-based one-shot FL methods. The contributions of this paper are:

- We propose a new one-shot FL framework named FedSD2C which proposes to share synthetic distillates instead of generating noisy data from inconsistent models for server-side training.
- To mitigate the potential of privacy leakage and reduce communication costs, we propose two techniques: distillate initialization with Fourier transform perturbation and distillate synthesis with a pre-trained Autoencoder.
- We conduct extensive experiments over various datasets and settings. The results demonstrate the effectiveness of the proposed method which achieves up to 2.7 × the performance of the best baseline.

2 Related Work

2.1 One-shot Federated Learning

One-shot federated learning was first proposed by [22], which introduces a method to aggregate a server model by distilling knowledge from an ensemble of client models using public datasets. FedKT [11] propose a hierarchical knowledge transfer framework, enabling various types of classification models. While their approaches demonstrate promising results, the requirement of public datasets which is inaccessible for privacy or transmission reasons limits their practical applications. To address the limitations associated with public datasets, DENSE [8] introduces a DFKD process utilizing an additional data generator trained on the ensemble model. Considering the challenge of high statistical heterogeneity, FedCVAE [9] proposes replacing the local training task with training conditional Variation Autoencoders. Furthermore, Co-Boosting [10] aims to enhance the performance of both the data generator and the ensemble model through a two-tier process. Despite these advancements, generating data through DFKD involves a two-tier information loss. Simultaneously, the inconsistency among client models due to data heterogeneity [15, 23, 24, 25], further degrades the quality of generated data, introducing label noise and thus limiting the performance of the server model. In this work, we tackle these problems from the perspective of sharing synthetic distillates. By utilizing Core-Set selection and pre-trained Autoencoders as distillers, our proposed methods distill diverse and informative data for server training.

2.2 Dataset Distillation in Federated learning

Dataset Distillation (DD) was first introduced by [26], aiming to distill the knowledge of datasets into synthetic data while preserving the performance of the model trained on it. Early dataset distillation methods are formulated as a bi-level optimization problems [27], where the outer loop optimize the synthetic data via gradient matching [28, 29, 30], distribution matching [31, 32] and performance matching [26], while the inner loop progressively trains a model on the synthetic data. Considering computational resources constraints, single-level optimization methods [33, 34] based on kernel ridge regression are proposed to decouple the bi-level optimization, thereby reducing training cost. These methods demonstrate comparable performance in non-complex datasets like CIFAR10 and are implemented in FL to tackle communication bottlenecks [35, 36], data heterogeneity [37, 38, 39] and one-shot FL [40, 36]. However, these methods require significant computational resources, making them impractical for edge devices with limited capability in FL. Additionally, they may struggle to effectively distill high-resolution datasets.

3 Methodology

3.1 Overview

We proposed FedSD2C to alleviate the two-tier information loss inherent in one-shot FL methods based on DFKD and account for data heterogeneity by synthetic distiller-distillate communication.

https://doi.org/10.52202/079017-3259

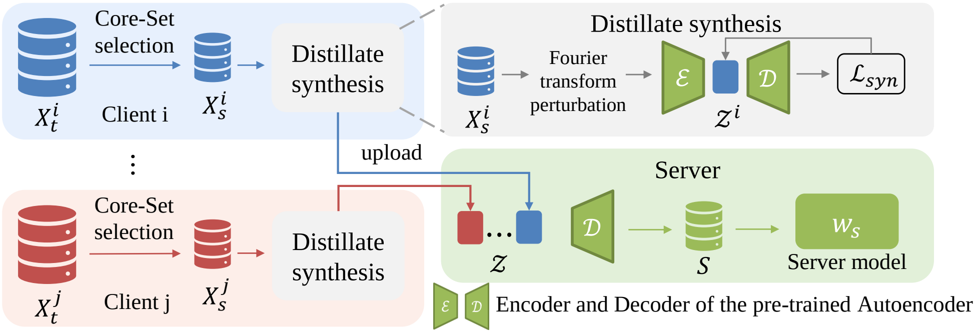


Figure 2: Framework of proposed FedSD2C.

The details of FedSD2C are described in Figure 2 and Algorithm 1. In the preparation phase of one-shot FL, the server distributed a pre-trained Autoencoder [21] as the distiller to each client. Subsequently, clients synthesize informative, privacy-enhanced, and communication-efficient distillates for server-side training. Specifically, to ensure the distillates fully encompass local information, clients first utilize a $\mathcal{V}$-information-based Core-Set selection method to extract diverse and informative Core-Set from their local data domains. Aiming to further reduce the communication costs and enhance the privacy of distillates, clients then perturb the Core-Set with Fourier transform for distillate initialization and employ the received pre-trained Autoencoder to optimize distillates in a compact latent space via $\mathcal{V}$-information alignment with the Core-Set. Finally, clients transmit the distillates, and the server decodes them using the pre-trained Autoencoder for training. We will now delve into the details of each component.

3.2 $\mathcal{V}$-information based Core-Set Selection

$\mathcal{V}$-information [20] was first proposed to measure the mutual information between X and Y constrained on predictive family V which is denoted as:

$$I_{\mathcal{V}}(X \to Y) = H_{\mathcal{V}}(Y|\varnothing) - H_{\mathcal{V}}(Y|X) \tag{1}$$

where $H_{\mathcal{V}}(Y|\varnothing)$ and $H_{\mathcal{V}}(Y|X)$ denote the predictive $\mathcal{V}$-entropy conditioned on $\varnothing$ or X.

Core-Set selection is a type of dataset distillation method that focuses on preserving a subset of the original training dataset containing only valuable or representative samples. The objective is to enable models trained on this subset to achieve performance similar to those trained on the entire dataset. Typically, this is achieved by minimizing certain criteria, such as data distribution [41]. In our case, the emphasis lies on maximizing the diversity and information content of the subset. Therefore, in the context of $\mathcal{V}$-information, Core-Set selection can be reformulated as:

$$(X_s, Y_s) = \arg\max_{X,Y} I_{\mathcal{V}}(X_t \to Y_t) \tag{2}$$

where X_t and Y_t denotes the images and labels in the original datasets and (X_s, Y_s) denotes the selected Core-Set. The intuition behind this equation is that Core-Set should include sufficient information and provide a concise representation corresponding to original datasets, constrained by observers $\mathcal{V}$.

Inspired by [42], we maximize the $\mathcal{V}$-information of the Core-Set from two levels. First, we identify the most informative image segments within each image by evaluating patches extracted at various scales from each image. Second, we select the top-ipc with the highest $\mathcal{V}$-information for each class to construct the final Core-Set. The algorithm description can be found at Appendix A. Since a model pre-trained on original datasets can serve as an optimal observer for approximating the $\mathcal{V}$-information [42], we proposed using local pre-trained models as the observer models (predictive family $\mathcal{V}$) to conduct $\mathcal{V}$-information-based Core-Set selection on local datasets. Given that pre-trained local models are commonly present in one-shot FL [8, 7], their utilization does not violate the practicality of our approach.

3.3 Distilling Core-Set into Distillates with Pre-trained Autoencoders

In this section, we will describe how to synthesize informative, privacy-enhanced, and communication-efficient distillates using pre-trained Autoencoders. Although Core-Set significantly reduces dataset

https://doi.org/10.52202/079017-3259

size, communication costs remain a challenge for high-resolution data in real-world applications. Moreover, while sharing Core-Set provides consistent resistance to membership inference attacks [43], transmitting patches may still risk exposing sensitive information of original images. Therefore, further enhancing the privacy of shared data is essential. To alleviate these concerns, we propose two novel techniques: 1) distillate initialization with Fourier transform perturbation, which alters the amplitude components of the Core-Set samples to enhance privacy while retaining semantic content; and 2) distillate synthesis with pre-trained Autoencoders which act as the distillers. The pre-trained Autoencoder converts the perturbed samples into distillates by optimizing their $\mathcal{V}$-information to be as close as possible to the original Core-Set samples, minimizing information loss. We will now discuss each component in detail below. The process is described in Algorithm 1.

Distillate initialization with Fourier transform perturbation. A typical privacy-enhanced technique for sharing synthetic data is adding noise [44]. Although this approach can blur the visual information of the synthetic data, it can also destroy important semantic information and seriously degrade model performance. Our approach leverages a well-known property of the Fourier transform: the phase component of the Fourier spectrum encodes high-level semantic information, whereas the amplitude component captures low-level details [45, 46]. Inspired by this, we propose a novel privacy-enhanced method that perturbs the amplitude components in Core-Set samples through the Fourier transform to reduce visual information while preserving semantic information. Given an image x, its Fourier transform can be formulated as:

$$\mathcal{F}(x) = \mathcal{A}(x) \times e^{-j \times \mathcal{P}(x)} \tag{3}$$

where $\mathcal{A}(x), \mathcal{P}(x)$ depict the amplitude and phase components respectively. We then perturb the amplitude information via linearly interpolating:

$$\hat{\mathcal{A}}(x) = (1 - \lambda)\mathcal{A}(x) + \lambda \mathcal{A}(x^*) \tag{4}$$

where the λ is a scaling coefficient and x^* can be other images or random noise. Then, we combine the perturbed amplitude spectrums with the original phase component to generate the perturbed Core-Set sample:

$$x = \mathcal{F}^{-1}(\hat{\mathcal{A}}(x) \times e^{-j \times \mathcal{P}(x)}) \tag{5}$$

where $\mathcal{F}^{-1}(x)$ defines the inverse Fourier transform which can be calculated with the FFT algorithm [47] effectively.

Distillate synthesis with pre-trained Autoencoders. The Fourier transform perturbation serves to protect visual privacy but also compromises the realism of images, resulting in inconsistent $\mathcal{V}$-information between the perturbed Core-Set and the original Core-Set. To address this issue, we propose optimizing the alignment of the $\mathcal{V}$-information between the perturbed Core-Set and the original Core-Set to reconstruct key information. One straightforward and efficient approach is to optimize the perturbed Core-Set in pixel space [48, 49]. However, this method can be prone to overfitting into high-frequency patterns that only match the observer model [50]. Such overfitting is detrimental to training a global model due to inconsistent local models caused by data heterogeneity. Leveraging the powerful priors obtained from large-scale datasets, a pre-trained Autoencoder can decode latent representations into generalizable images. As a result, optimizing in the latent space acts as a regularization method that encourages synthetic data to be more generalizable, thereby making pre-trained Autoencoder an ideal distiller for local Core-Sets distillation. Furthermore, compact latent representations can reduce communication costs and mitigate privacy leakage if the latent is intercepted by attackers. Consequently, we employ a pre-trained Autoencoder on the client to distill the Core-Set into informative, privacy-enhanced, and communication-efficient distillates, and transmit them to the server for model training. On the server side, they are decoded by the decoder and are expected to maintain similar $\mathcal{V}$-information to the original Core-Set from the perspective of the observer model while remaining visually unidentifiable.

In specific, in the preparation phase of one-shot FL, the server first distributes a pre-trained Autoencoder to n clients, denoted as $\mathcal{E}$ and $\mathcal{D}$ for encoder and decoder, respectively. Each client i then conducts $\mathcal{V}$-information-based selection to construct a Core-Set $(X_s^i, Y_s^i), i = 1, 2 \cdots, n$ with diverse information regarding the original local datasets. Subsequently, client i learns a latent set $\mathcal{Z}^i = \{z_j\}_{j=1}^{|\mathcal{Z}^i|}$ initialized by $\{\mathcal{E}(x_j^i)\}_{j=1}^{|X_s^i|}, x_j^i \in X_s^i$ which is perturbed with Fourier transform, such that the $\{\mathcal{D}(z_j^i)\}_{j=1}^{|\mathcal{Z}^i|}$ is as close as possible to the corresponding data in the Core-Set:

$$\underset{\mathcal{Z}^i}{\arg\min} \left\| I_{\mathcal{V}^i}(X_s^i \to Y_s^i) - I_{\mathcal{V}^i}(\{\mathcal{D}(z_j^i)\}_{j=1}^{|\mathcal{Z}^i|} \to Y_s^i) \right\|^2 \tag{6}$$

Algorithm 1 One-shot Federated Learning via Synthetic Distiller-Distillate Communication

```
Require: Client local model $f^i(h^i(\cdot))$, pre-trained VAE $(\mathcal{D}, \mathcal{E})$, Server model parameter $\theta$, number of clients $n$, training iterations of local synthesis $T_{syn}$, training iterations of server model $T_{trn}$, learning rate $\eta_{syn}$ and $\eta_{trn}$
Server distributes pre-trained VAE $(\mathcal{D}, \mathcal{E})$ to $n$ clients.
for each client $i = 1, \cdots, n$ do
    $(X_s^i, Y_s^i) \leftarrow \text{CoresetSelection}(i)$        ▷ See Algorithm 2 in Appendix
    for each $x_j^i \in X_s^i$ do
        Perturb $x_j^i$ via Equations (3), (4) and (5)
    end for
    Initialize latent set $\mathcal{Z}^i = \{\mathcal{E}(x_j^i)\}_{j=1}^{|X_s^i|}$ using pre-trained VAE decoder $\mathcal{E}$
    for $t = 1, \cdots T_{syn}$ do
        for mini-batch $(z^i, x_s^i) \in (\mathcal{Z}^i, X_s^i)$ do
            Compute synthetic loss $\mathcal{L}_{syn}$ based on Equation (7)
            $z^i \leftarrow z^i - \eta_{syn} \nabla_{z^i} \mathcal{L}_{syn}$
        end for
    end for
    Generate soft label for each synthetic latents $Y_s^i = \{f^i(h^i(\mathcal{D}(z_j^i)))\}_{j=1}^{|\mathcal{Z}^i|}$
    Transmit $S^i = (\mathcal{Z}^i, Y_s^i)$ to the server
end for
Combine client synthetic data into $S = (\mathcal{Z}, Y_s) = (\mathcal{Z}^1 \cup \cdots \cup \mathcal{Z}^n, Y_s^1 \cup \cdots \cup Y_s^n)$
for $t = 1, \cdots, T_{trn}$ do
    for mini-batch $(z, y) \in (\mathcal{Z}, Y_s)$ do
        Compute $\mathcal{L}_{trn}$ based on Equation (8)
        $\theta \leftarrow \theta - \eta_{trn} \nabla_\theta \mathcal{L}_{trn}$
    end for
end for
```

As Core-Set selection employs the local pre-trained models as the observer models $\mathcal{V}^i$, the reformulated Equation (6) and objective function can be formulated as:

$$\begin{aligned} &\arg\min_{z^i} \left\| h_i(\mathcal{D}(z^i)) - h_i(x^i) \right\|^2 \\ \mathcal{L}_{syn} = &\left\| \frac{1}{N} \sum_{j=1}^{N} h_i(\mathcal{D}(z_j^i)) - \frac{1}{N} \sum_{j=1}^{N} h_i(x_j^i) \right\|^2 \end{aligned} \tag{7}$$

where $h_i(\cdot)$ denotes the feature extractor of pre-trained local model of client i, and x_j^i and z_j^i are paired. By minimizing $\mathcal{L}_{syn}$, we synthesize a set of latent variables $\mathcal{Z}^i = \{z_j\}_{j=1}^{|\mathcal{Z}^i|}$ that contains diverse information of local data domains.

Finally, clients transmit the latent set $\mathcal{Z}^i$ along with the corresponding soft label Y_s^i predicted by local models to the server. The server combines the synthetic local distillates from each client $S = (\mathcal{Z}, Y_s)$. It then uses decoder $\mathcal{D}$ to reconstruct the images from data $(z, y) \in (\mathcal{Z}, Y_s)$ and distills the knowledge by minimizing the following objective function:

$$\mathcal{L}_{trn} = \sum_{(z,y) \in (\mathcal{Z}, Y_s)} KL(f(h(\mathcal{D}(z))), y) \tag{8}$$

where $f(h(\cdot))$ denotes the server model. By minimizing the KL loss, we can transfer the local knowledge in the distillate to the server model.

Discussion on privacy. We first consider whether an attacker can train a performant model with the intercepted distilled data and labels during transmission [40, 9]. Because the attacker cannot know that the distilled data is encoded by VAE, nor can the attacker access the pre-trained VAE encoder, which can be easily achieved by being predefined offline or via encryption, it is hard for the attacker to reproduce an effective model. For model inversion and membership inference attacks, according to [51], there has been no prior research has performed these attacks solely using distilled data and

Table 1: Accuracy of different one-shot FL methods over three datasets with ConvNet and ResNet-18. We vary the $\alpha = \{0.1, 0.3, 0.5\}$ to simulate different levels of data heterogeneity for Tiny-ImageNet and Imagenette and use pre-defined splits for OpenImage. "Central" means that clients send all their local data to the server for centralized training, representing the upper bound of model performance.

Model	Methods	ImageNette			Tiny-ImageNet			OpenImage
		$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	-
ConvNet	Central	89.60			49.73			33.61
	FedAVG	10.68±0.23	10.04±0.10	9.83±0.27	-	-	-	3.08±0.17
	F-DAFL	44.95±0.72	52.23±0.23	58.34±0.55	5.25±0.41	8.89±0.61	10.28±0.10	3.36±0.56
	DENSE	42.09±0.68	48.64±1.91	54.74±0.75	11.45±0.08	14.69±0.48	15.15±0.22	7.00±0.84
	Co-Boosting	39.36±0.70	56.15±1.33	**58.60±1.02**	6.66±0.35	9.81±0.26	10.75±0.11	13.59±0.98
	FedSD2C	**50.68±0.20**	**57.89±0.96**	58.17±0.51	**20.73±0.12**	**23.53±0.18**	**24.10±0.30**	**23.00±0.24**
ResNet-18	Central	90.00			61.98			34.17
	FedAVG	9.86±0.13	10.06±0.20	10.76±0.35	-	-	-	1.68±0.16
	F-DAFL	37.86±0.38	39.52±0.46	46.06±0.16	7.91±0.22	12.30±0.36	13.31±0.56	12.75±0.14
	DENSE	38.37±0.36	47.85±2.17	49.78±2.11	8.88±0.23	13.05±0.36	17.24±0.43	14.85±0.62
	Co-Boosting	27.06±0.61	28.53±0.86	30.53±1.12	10.29±0.43	14.35±0.93	16.39±0.59	9.52±1.52
	FedSD2C	**47.52±0.51**	**53.69±0.17**	**55.90±0.53**	**26.83±0.10**	**29.92±0.37**	**31.66±0.85**	**22.69±0.14**

labels and previous works [52, 36, 35, 43] have also revealed the advantage of dataset distillation in this regard. Therefore, we employ the synthetic data as the reconstructed samples for the evaluation of model inversion attacks. Furthermore, we compare our proposed Fourier transform perturbation with other privacy-enhanced techniques, including adding random noise to synthetic samples and data augment [53]. Experimental results can be found in Section 4.3.1.

4 Experiments

4.1 Experimental Setup

Datasets and partitions. We conduct experiments on three real-world image datasets with different ranges of resolution including Tiny-ImageNet [54], ImageNette [55], and OpenImage [56]. Tiny-ImageNet contains 10000 images of 64×64 resolution across 200 classes. ImageNette is a widely used subset of 10 classes from ImageNet-1K [57] with 9469 color images, resized to 128×128. OpenImage is a large-scale real-world vision dataset with over 9 million images of 256×256 resolution. To simulate data heterogeneity in real-world applications of one-shot FL, we use Dirichlet distribution to generate non-IID data to generate non-IID local data, as in [58] for Tiny-ImageNet and ImageNette. Specifically, for client i, we sample p_k^i $Dir(\alpha)$ to allocate a p_k^i proportion of class k to client i. The parameter α controls the degree of data heterogeneity, with smaller α indicating severe data heterogeneity. The α is set to 0.1 by default unless otherwise stated. For OpenImage, we randomly choose n real-world clients from FedScale [59] and use their corresponding test sets to form global sets. We set the default number of clients n to 10, unless otherwise specified.

Baseline methods and Configurations. We compare our proposed FedSD2C with existing methods: FedAVG [1], DENSE [8] and Co-Boosting [10]. Following [8, 10], we also introduce DAFL [60] with one-shot FL settings, denoted as F-DAFL. We use two different model architectures: ConvNet [28] and ResNet-18 [61] for all methods. In FedSD2C, the image per class ipc is set to 50 for Tiny-ImageNet and ImageNette, and 10 for OpenImage. We set the Fourier transform coefficient $\lambda = 0.8$ and use a public pre-trained Autoencoder from Stable Diffusion [62] by default for all tasks. For distillate synthesis, we set $T_{syn} = 50$, $\eta_{syn} = 0.1$ by default. η_{trn} is set to 0.2 for Tiny-ImageNet and 0.02 for ImageNette and OpenImage. More experimental details can be found in the Appendix.

4.2 Evaluation Results

To evaluate the effectiveness of our method, we conduct experiments under various non-IID settings with $\alpha = \{0.1, 0.3, 0.5\}$ for Tiny-ImageNet and Imagenette and pre-defined splits [59] for OpenImage. As illustrated in Table 1, our proposed FedSD2C surpasses all other methods in most settings. In particular, under extreme data heterogeneity($\alpha = 0.1$), FedSD2C achieves up to 1.3×, 2.6×, and 1.8× the accuracy of the best baseline on ImageNette, Tiny-ImageNet and OpenImage, respectively. This superior performance is attributed to FedSD2C's approach of sharing synthetic

Table 2: Accuracy, PSNR and SSIM of FedSD2C combining different privacy-enhanced techniques. $Laplace$ and $Gaussian$ indicate adding corresponding noise into synthetic distillates without Fourier transform initialization. FedMix denotes averaging two real samples from Core-Set to synthesize data. "-" indicates no privacy-enhanced technique is combined.

Privacy-preserving	ImageNette				Tiny-ImageNet			
techniques	ConvNet↑	ResNet-18↑	PSNR↓	SSIM↓	ConvNet↑	ResNet-18↑	PSNR↓	SSIM↓
-	51.87	51.82	-	-	22.62	28.29	-	-
Ours($\lambda = 0.1$)	51.26	50.55	23.48	73.20	22.03	28.22	20.54	54.89
Ours($\lambda = 0.5$)	51.36	48.97	19.97	64.23	21.77	28.09	18.06	44.18
Ours($\lambda = 0.8$)	50.68	47.52	16.42	50.80	20.85	26.83	16.95	35.89
$Laplace(s = 0.2, p = 0.1)$	48.61	45.25	24.02	81.66	21.50	27.48	22.25	73.09
$Gaussian(s = 0.2, p = 0.1)$	48.31	46.70	24.82	85.89	21.48	27.51	23.38	78.90
$Laplace(s = 0.2, p = 0.2)$	45.61	38.01	20.05	73.13	19.32	23.66	19.99	64.51
$Gaussian(s = 0.2, p = 0.2)$	45.81	38.09	20.30	76.11	19.32	23.52	20.35	68.56
FedMix	41.86	37.76	16.88	58.93	13.86	16.26	16.43	56.91

distillates rather than inconsistent local models, thereby mitigating the impact of data heterogeneity. Moreover, FedSD2C demonstrates the independence from model structures. In contrast, other methods struggle to adapt to different model structures and complex datasets. For instance, at $\alpha = 0.5$, Co-Boosting with ResNet-18 achieves only half the accuracy of ConvNet on ImageNette, whereas FedSD2C maintains consistent performance. This discrepancy arises because differences in model capacity affect their ability to condense local knowledge and the two-tier information loss during data generation increases the difficulty of transferring local knowledge to the server model, resulting in poor robustness to complex datasets and varied networks. In contrast, the shared distillates in FedSD2C are synthesized through end-to-end local distillation, mitigating information loss during knowledge transfer.

4.3 Analysis of Our Method

4.3.1 Privacy Evaluation

For privacy evaluation, we consider an *honest-but-curious* server attempting to reconstruct client data from distillates. We compare our proposed FedSD2C with other privacy-enhanced techniques for sharing synthetic data, including adding random noise [44, 63] and FedMix [53]. In the random noise approach, we incorporate it into FedSD2C by removing the Fourier transform perturbation and instead directly using Core-Set samples for initialization. We then add random noise to the synthetic distillates before transmitting them to the server, following the methods in [44, 63]. Specifically, Given latent z, a perturbation coefficient p, randomly generated noise e and its scale parameter s, the data to be shared is formulated as $z = (1 - p)z + e \times s$. FedMix [53] proposes using linear interpolation of real samples to preserve privacy. In this approach, we synthesize data by averaging each two real samples from the Core-Set. For our proposed Fourier transform perturbation, we vary the $\lambda = \{0.1, 0.5, 0.8\}$ and observe the variations in performance and privacy protection. To quantitatively evaluate the privacy protection of the synthetic data, we employ the Peak Signal-to-Noise Ratio (PSNR) and Structure Similarity Index Measure (SSIM). A higher PSNR or SSIM value indicates greater similarity between the synthetic samples and the original samples, which implies more severe privacy leakage. We calculate the average PSNR and SSIM values of all the synthetic samples.

As depicted in Table 2, although FedMix provides better privacy protection, as evidenced by lower PSNR and SSIM values, it comes at the expense of significant performance degradation. The application of random noise requires a delicate balance between performance and privacy protection. For example, with a perturbation coefficient $p = 0.2$, it offers similar privacy protection to that of FedMix, but the performance drops approximately 10% compared to $p = 0.1$. However, the $p = 0.1$ setting increases the risk of privacy leakage. In comparison, the synthetic distillates generated by our proposed FedSD2C achieve comparable PSNR values with them. This suggests that our proposed Fourier transform perturbation offers effective privacy protection for the real data sample. Furthermore, FedSD2C consistently outperforms other methods in terms of accuracy while maintaining a minimal performance degradation compared to no privacy protection techniques. This indicates that FedSD2C strikes a balance between privacy preservation and performance. We also

Table 3: Comparison of communication costs and accuracy at $\alpha = 0.1$ with ResNet-18. Results highlighted in **bold** represent outcomes with default ipc settings. Acc. and Comm. denote accuracy and communication costs, respectively.

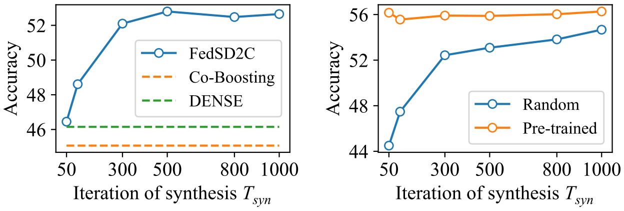
Method	ipc	ImageNette		Tiny-ImageNet		ipc	OpenImage	
		Acc.	Comm.	Acc.	Comm.		Acc.	Comm.
DENSE	-	38.37	44MB	8.88	44MB	-	14.85	44MB
Co-Boosting	-	27.06	44MB	10.29	44MB	-	7.00	44MB
FedSD2C w/o AE	1	19.90	0.48MB	3.60	2.2MB	1	12.89	2.6MB
FedSD2C	20	43.10	0.23MB	23.23	1.1MB	5	20.73	1.6MB
	50	**50.68**	**0.5MB**	**26.83**	**2.1MB**	**10**	**22.69**	**2.0MB**
	80	56.13	0.73MB	27.71	2.9MB	15	23.49	2.7MB

perform membership inference attacks on FedSD2C and other methods, please refer to Appendix C.2 for more details.

4.3.2 Scalability of Communication Efficiency

By employing Core-Set selection and communication-efficient distillate communication, our FedSD2C condenses local data to mere MBs, while the model trained on these condensed data exhibits comparable performance, as shown in Table 3. Specifically, the communication costs of FedSD2C for sharing synthetic distillate is at most 4% of that of sharing model. Notably, we exclude the communication costs of sending and receiving pre-trained Autoencoder, as this can be pre-defined offline, allowing for the reuse of multiple one-shot FL tasks. Given the considerable capacity for communication costs, we further investigate the scalability of communication efficiency and performance. Our experimental results demonstrate that increased data transmission enhances the diversity of compressed data, leading to further improvements in performance. By increasing ipc from 20 to 80, the accuracy boosts by 13.03% and 4.48% on ImageNette and Tiny-ImageNet respectively. Additionally, we compare FedSD2C without Autoencoder at equivalent communication costs, where the absence of synthesized images limits performance to at best half of the default setting. The communication-efficiency of FedSD2C highlights its practicality in real-world applications.

4.3.3 Impact of Pre-trained Autoencoder



(a) COVID-FL

(b) ImageNette

Figure 3: (a) Experiments on the medical image data domain. Adopting pre-trained Autoencoders on other data domains can reduce performance. However, this can be mitigated by increasing T_{syn}. (b) Experiments of FedSD2C with randomly initialized downsampling and upsampling modules (blue line) compared to pre-trained Autoencoders (orange line) on ImageNette. Without pre-trained knowledge, FedSD2C requires a higher T_{syn} for distillate synthesis but can still achieve comparable results. ResNet-18 is used for both experiments.

Pre-trained Autoencoders are typically trained on natural data domains [64], while practical applications of federated learning often involve a broader range of domains, such as medical images. This raises the question of whether pre-trained Autoencoders remain effective when applied to a different domain and whether our proposed FedSD2C can adapt to these differences. To investigate this, we evaluate performance using a medical dataset COVID-FL [65].

As shown in Figure 3a, using the default synthesis iteration of $T_{syn} = 50$ yields suboptimal results. By increasing T_{syn} to 1000, the performance improves and then stabilizes. This observation suggests that the pre-trained knowledge of Autoencoders may influence the speed of distillate synthesis convergence. To further validate this, we replace the encoder and decoder of pre-trained Autoencoders with randomly initialized downsampling and upsampling modules. We vary T_{syn} from 50 to 1000 and compare

Table 4: Accuracy on Tiny-ImageNet with different client amounts.

Number of clients	α	F-DAFL	DENSE	Co-Boosting	FedSD2C
n=20	$\alpha = 0.1$	4.97	11.36	7.37	**21.92**
	$\alpha = 0.3$	7.66	13.69	9.95	**22.00**
	$\alpha = 0.5$	10.01	14.86	10.45	**22.87**
n=50	$\alpha = 0.1$	3.99	8.32	7.05	**21.48**
	$\alpha = 0.3$	5.92	12.25	8.80	**22.38**
	$\alpha = 0.5$	6.94	13.06	8.90	**22.58**
n=100	$\alpha = 0.1$	3.12	7.87	4.27	**20.34**
	$\alpha = 0.3$	5.26	10.22	7.15	**21.86**
	$\alpha = 0.5$	6.30	11.49	8.32	**21.76**

its performance with employing pre-trained Autoencoders on ImageNette, setting $ipc = 80$ for better illustration. Figure 3b demonstrates that as T_{syn} increases, the performance of FedSD2C with randomly initialized modules improves progressively, eventually matching the performance of FedSD2C with pre-trained Autoencoders. In summary, while pre-trained knowledge can enhance convergence rate, FedSD2C can achieve comparable performance by adjusting T_{syn}, demonstrating its adaptability across domains.

4.3.4 Impact of Client Scales

As practical FL deployments often involve participating clients [59], we evaluate our FedSD2C with various numbers of clients $n = \{20, 50, 100\}$ and maintain consistent communication budget by setting $ipc = \{40, 20, 10\}$, respectively. We compare these methods under on Tiny-ImageNet with data heterogeneity $\alpha = \{0.1, 0.3, 0.5\}$ for partitions and employ ConvNet. As depicted in Table 4, FedSD2C consistently achieves the highest accuracy as the number of clients increases. Moreover, FedSD2C demonstrates greater robustness to the number of participants. Specifically, as the number of participants changes, the accuracy of FedSD2C fluctuates within only 1%. In contrast, the accuracy of F-DAFL, DENSE, and Co-Boosting dropped by up to 4.71%, 3.49%, and 3.10%, respectively under different settings. This further validates the utility of sharing synthesized distillates in real-world one-shot FL applications.

5 Limitations

The local distillation process introduces additional computational overhead. While Core-Set selection requires no training and the distillate synthesis process only requires 50 iterations with a speed of 0.4s/per image on RTX3090, it still imposes higher resource requirements on the local device compared to the method of sharing model. One direction worth exploring is to integrate with the model market [66] to enable clients to synthesize distillates once for permanent use.

6 Conclusion

In this paper, we propose a new one-shot FL framework driven by distiller-distillate communication, denoted as FedSD2C, to alleviate the information loss of knowledge transfer and impacts of data heterogeneity. FedSD2C compresses the local data into Core-Set with $\mathcal{V}$-information and employs a pre-trained Autoencoder as the distiller to distill informative, communication-efficient, and privacy-enhanced distillates from Core-Set. Moreover, We discuss FedSD2C's resistance to attackers intercepting distillate communications and attacks from honest-but-curious servers and introduce Fourier transform perturbation to further minimize the risk of privacy leakage. Empirical results validate the effectiveness of FedSD2C in transferring local knowledge to the server in one-shot FL while balancing communication efficiency and privacy protection.

Acknowledgment

This project is supported by the National Research Foundation, Singapore, under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-023).

References

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

[2] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[3] Peter Kairouz, H Brendan Mcmahan, Brendan Avent, AurÉLien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, and Others. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[4] Junyuan Zhang, Shuang Zeng, Miao Zhang, Runxi Wang, Feifei Wang, Yuyin Zhou, Paul Pu Liang, and Liangqiong Qu. Flhetbench: Benchmarking device and state heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12098–12108, June 2024.

[5] Derui Wang, Chaoran Li, Sheng Wen, Surya Nepal, and Yang Xiang. Man-in-the-middle attacks against machine learning classifiers via malicious generative models. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2074–2087, 2020.

[6] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16337–16346, 2021.

[7] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arxiv preprint arxiv:1902.11175*, 2019.

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

[9] Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. Data-free one-shot federated learning under very high statistical heterogeneity. In *ICLR*, 2022.

[10] Rong Dai, Yonggang Zhang, Ang Li, Tongliang Liu, Xun Yang, and Bo Han. Enhancing one-shot federated learning through data and ensemble co-boosting. *arxiv preprint arxiv:2402.15070*, 2024.

[11] Qinbin Li, Bingsheng He, and Dawn Song. Practical one-shot federated learning for cross-silo setting. *arxiv preprint arxiv:2010.01017*, 2020.

[12] Xingyi Yang, Daquan Zhou, Songhua Liu, Jingwen Ye, and Xinchao Wang. Deep model reassembly. *Advances in neural information processing systems*, 35:25739–25753, 2022.

[13] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pages 12878–12889. PMLR, 2021.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[15] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[16] Jiawei Shao, Fangzhao Wu, and Jun Zhang. Selective knowledge sharing for privacy-preserving federated distillation without a good teacher. *Nature Communications*, 15(1):349, 2024.

[17] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.

[18] Xingyi Yang, Jingwen Ye, and Xinchao Wang. Factorizing knowledge in neural networks. In *European Conference on Computer Vision*, pages 73–91. Springer, 2022.

[19] Shikang Yu, Jiachen Chen, Hu Han, and Shuqiang Jiang. Data-free knowledge distillation via feature exchange and activation region constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24266–24275, 2023.

https://doi.org/10.52202/079017-3259

[20] Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. *arxiv preprint arxiv:2002.10689*, 2020.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arxiv preprint arxiv:1312.6114*, 2013.

[22] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arxiv preprint arxiv:1902.11175*, 2019.

[23] Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10061–10071, 2022.

[24] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[25] Shuang Zeng, Pengxin Guo, Shuai Wang, Jianbo Wang, Yuyin Zhou, and Liangqiong Qu. Tackling data heterogeneity in federated learning via loss decomposition. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 707–717. Springer, 2024.

[26] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arxiv preprint arxiv:1811.10959*, 2018.

[27] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *PAMI*, 2023.

[28] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *ICLR*, 2021.

[29] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4750–4759, 2022.

[30] Ruonan Yu, Songhua Liu, Jingwen Ye, and Xinchao Wang. Teddy: Efficient large-scale dataset distillation via taylor-approximated matching. In *European Conference on Computer Vision*, pages 1–17. Springer, 2025.

[31] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *WACV*, pages 6514–6523, 2023.

[32] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12196–12205, 2022.

[33] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. *ICLR*, 2021.

[34] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198, 2021.

[35] Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. Feddm: Iterative distribution matching for communication-efficient federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16323–16332, 2023.

[36] Rui Song, Dai Liu, Dave Zhenyu Chen, Andreas Festag, Carsten Trinitis, Martin Schulz, and Alois Knoll. Federated learning via decentralized dataset distillation in resource-constrained edge environments. In *IJCNN*, pages 1–10. IEEE, 2023.

[37] Renjie Pi, Weizhong Zhang, Yueqi Xie, Jiahui Gao, Xiaoyu Wang, Sunghun Kim, and Qifeng Chen. Dynafed: Tackling client data heterogeneity with global dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12177–12186, 2023.

[38] Ping Liu, Xin Yu, and Joey Tianyi Zhou. Meta knowledge condensation for federated learning. *ICLR*, 2023.

[39] Jie Zhang, Xiaohua Qi, and Bo Zhao. Federated generative learning with foundation models. *arXiv preprint arXiv:2306.16064*, 2023.

[40] Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. Distilled one-shot federated learning. *arxiv preprint arxiv:2009.07999*, 2020.

[41] Max Welling. Herding dynamical weights to learn. In *International Conference on Machine Learning*, pages 1121–1128, 2009.

[42] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. on the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. *arxiv preprint arxiv:2312.03526*, 2023.

[43] Hanlin Lu, Changchang Liu, Ting He, Shiqiang Wang, and Kevin S Chan. Sharing models or coresets: A study based on membership inference attack. *arxiv preprint arxiv:2007.02977*, 2020.

[44] Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. Federated learning from pre-trained models: A contrastive learning approach. *Advances in Neural Information Processing Systems*, 35:19332–19344, 2022.

[45] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14383–14392, 2021.

[46] Leon N Piotrowski and Fergus W Campbell. A demonstration of the visual importance and flexibility of spatial-frequency amplitude and phase. *Perception*, 11(3):337–346, 1982.

[47] Henri J Nussbaumer and Henri J Nussbaumer. *the Fast Fourier Transform*. Springer, 1982.

[48] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8715–8724, 2020.

[49] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

[50] Jonas Geiping, Hartmut Bauermeister, Hannah DrÖGe, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[51] Jiawei Shao, Zijian Li, Wenqiang Sun, Tailin Zhou, Yuchang Sun, Lumin Liu, Zehong Lin, and Jun Zhang. A survey of what to share in federated learning: Perspectives on model utility, privacy leakage, and communication efficiency. *arxiv preprint arxiv:2307.10655*, 2023.

[52] Tian Dong, Bo Zhao, and Lingjuan Lyu. Privacy for free: How does dataset condensation help privacy? In *International Conference on Machine Learning*, pages 5378–5396. PMLR, 2022.

[53] Tehrim Yoon, Sumin Shin, Sung Ju Hwang, and Eunho Yang. Fedmix: Approximation of mixup under mean augmented federated learning. *arxiv preprint arxiv:2107.00233*, 2021.

[54] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *Cs 231n*, 7(7):3, 2015.

[55] Imagenette. `https://github.com/fastai/imagenette`.

[56] Google Open Images Dataset. `https://storage.googleapis.com/openimages/web/index.html`.

[57] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[58] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arxiv preprint arxiv:1909.06335*, 2019.

[59] Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury. Fedscale: Benchmarking model and system performance of federated learning at scale. In *International Conference on Machine Learning*, pages 11814–11827. PMLR, 2022.

[60] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3514–3522, 2019.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

https://doi.org/10.52202/079017-3259

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

[63] Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. An upload-efficient scheme for transferring knowledge from a server-side pre-trained generator to clients in heterogeneous federated learning. *arxiv preprint arxiv:2403.15760*, 2024.

[64] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, and Others. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[65] Rui Yan, Liangqiong Qu, Qingyue Wei, Shih-Cheng Huang, Liyue Shen, Daniel Rubin, Lei Xing, and Yuyin Zhou. Label-efficient self-supervised federated learning for tackling data heterogeneity in medical imaging. *IEEE Transactions on Medical Imaging*, 2023.

[66] Manasi Vartak, Harihar Subramanyam, Wei-En Lee, Srinidhi Viswanathan, Saadiyah Husnoo, Samuel Madden, and Matei Zaharia. Modeldb: A system for machine learning model management. In *Proceedings of the Workshop on Human-in-the-Loop Data Analytics*, pages 1–3, 2016.

[67] Alex Krizhevsky, Geoffrey Hinton, and Others. Learning multiple layers of features from tiny images. 2009.

[68] Michael Aerni, Jie Zhang, and Florian TramÈR. Evaluations of machine learning privacy defenses are misleading. *arxiv preprint arxiv:2404.17399*, 2024.

[69] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

A $\mathcal{V}$-information based Core-Set Selection Description

Algorithm S1 $\mathcal{V}$-information-based Core-Set Selection

```
Require: Client local dataset $(X_t, Y_t)$ Client local feature extractor $h$, image per class $ipc$
$X'_T \leftarrow \emptyset$
Level 1: identifying the most informative image segments
for each $x, y \in (X_t, Y_t)$ do
    $\{x^k\}_{k=1}^{K} \leftarrow$ extract $K$ patches with multiple scales from $x$.
    for $k = 1$ to $K$ do
        Calculate $\mathcal{V}$-information score $s^k \leftarrow -\mathcal{L}(h(x^k), y)$
    end for
    select $x'$ with highest score $s'$
    $X'_T \bigcup \{(x', y, s')\}$
end for
Level 2: select the top-$ipc$ patches with highest $\mathcal{V}$-information
$X_s, Y_s \leftarrow \emptyset$
for each class $c \in Y_t$ do
    if size of $\{(x'_j, y_j, s'_j)\}, y_j = c \geq ipc$ then
        select top-$ipc$ $\{(x'_j, y_j)\}_{j=1}^{ipc}$ via $s'_j$
        $X_s \bigcup \{x'_j\}_{j=1}^{ipc}, Y_s \bigcup \{y_j\}_{j=1}^{ipc}$
    end if
end for
return $(X_s, Y_s)$
```

B More Experimental Details

We use the SGD optimizer with momentum=0.9, learning rate=0.01 and weight decay=0.0001 for clients' local training. The batch size is set to 128 and local epoch is 200. For all generation based methods, we set the resolution of the generated images to 64×64, 128×128 and 256×256 for Tiny-ImageNet, ImageNette and OpenImage, the number of generated images in each batch is 128, and the learning rate of the generator is 0.001, the latent dimension is 256, iteration for training generator is 30, using Adam for optimization. The server model is optimized with SGD with momentum 0.9, the learning rate is 0.01, and the training epochs are 200. The synthesized batch size and server model training batch size is both 128. In DENSE, we set $\lambda_1 = 1$ for BN loss and $\lambda_2 = 0.5$ for diversity loss. In Co-Boosting, the perturbation strength is set to $\epsilon = 8/255$ and the step size $\mu = 0.1/n$. In Core-Set selection stage of FedSD2C, for each image x_i, we employ the `torchvision.transform.RandomResizeCrop` K times to generate a collection of patches. For patch size, we set the scale=(0.08, 1.0), which is to collect diverse image patches. Following [42], we employ ConvNet-4 for Tiny-ImageNet, ConvNet-5 for ImageNette and ConvNet-6 for OpenImage. All methods are implemented with Pytorch and conducted on GeForce RTX 3090.

Details of t-SNE plots in Figure 1 We randomly select a client and five classes from its local dataset (Tiny-ImageNet) and employs its local model (ResNet-18) to extract features. The feature is extracted from the final layer (before the classifier). We then use t-SNE plots to illustrate the feature distribution.

C Additional Experiments

C.1 Additional datasets

CIFAR-10. We set the resolution of generated images to 32×32 for CIFAR-10 [67] and keep all the other settings the same. As shown in Table S1, there is an initial performance discrepancy at the standard setting of $ipc = 50$. This occurs because our method prioritizes efficiency with large datasets rather than low-resolution ones. However, upon increasing the amount of synthetic data ($ipc = 500$), our method can still achieve comparable results.

Table S1: Performance on CIFAR-10 with ResNet-18.

Method	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	Comm.
DENSE	47.75	54.53	66.05	44MB
Co-Boosting	53.33	61.75	**68.99**	44MB
FedSD2C ($ipc = 50$)	44.72	48.23	51.14	0.03MB
FedSD2C ($ipc = 500$)	**56.95**	**62.37**	65.06	0.24MB

COVID-FL. We crop the images of COVID-FL [65] into 128×128, set the $T_{syn} = 1000$ as in Section 4.3.3 and keep all the other settings the same. The results in Table S2 indicate that our FedSD2C still acheive better results compared to DENSE and Co-Boosting.

Table S2: Performance on COVID-FL datasets with ResNet-18.

Method	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
DENSE	46.38	57.55	62.83
Co-Boosting	45.07	60.27	65.81
FedSD2C	52.65	62.50	66.68

C.2 Membership Inference Attack

To further validate the effectiveness of our method, we employ an improved version of LiRA [68] to conduct Membership Inference Attacks on our methods. When attacking each client, for FedSD2C, we use the distillates uploaded by the client to train a new model and conduct membership inference attacks on that model. For the sharing model-based methods, we perform membership inference attacks on the models uploaded by the clients. The client model is ResNet-18 with $\alpha = 0.1, ipc = 50$. We set the raw images of Core-Set as the canary (target data x), as this is the most serious case of our methods. The results confirm that our approach does not introduce more privacy risk than the sharing model-based approach, even for the most vulnerable targets.

Table S3: Membership Inference Attack.

Method	TPR@FPR=0.1%
Sharing model-based methods (DENSE, Co-Boosting)	22.81
FedSD2C	20.13

C.3 Wavelet transform perturbation

We explore the use of wavelet transforms to replace the Fourier transforms during Fourier transform perturbations. We use ResNet-18 on Tiny-ImageNet with $\alpha = 0.1, ipc = 50$. The results indicate that Wavelet Transform offers greater scalability in privacy protection. By increasing λ, the PSNR/SSIM can be reduced to as low as 12.90/15.30. When the accuracy is comparable to that of Fourier transform (Wavelet $\lambda = 0.5$ vs. Fourier $\lambda = 0.8$), the PSNR and SSIM of Wavelet transform is lower.

Table S4: Comparison between Wavelet transform and Fourier transform.

	Acc.	PSNR	SSIM
Wavelet($\lambda = 0.1$)	28.05	18.86	44.76
Fourier($\lambda = 0.1$)	28.22	20.54	51.50
Wavelet($\lambda = 0.5$)	26.91	15.22	27.34
Fourier($\lambda = 0.5$)	28.09	18.06	43.26
Wavelet($\lambda = 0.8$)	26.06	12.90	15.30
Fourier($\lambda = 0.8$)	26.83	16.95	35.89

C.4 Ablation study on Core-Set selection

In this section, we perform ablation experiments to explore the significance of $\mathcal{V}$-information Core-Set selection. We use ResNet-18 with $\alpha = 0.1, ipc = 50$. Compared with $\mathcal{V}$-information Core-Set selection, the accuracy of random selection decreased by 3.5 and 5.46 on Tiny-ImageNet and ImageNette, respectively. We also report the performance of uploading Core-Set directly, which achieve the best performance. However, without distillate synthesis, it will increase the cost of communication and the risk of privacy leakage.

Table S5: Performance of different selection strategy. Core-Set denotes that clients directly upload their local Core-Set, which leads to privacy issue. FedSD2C w/ random selection denotes replacing $\mathcal{V}$-information-based Core-Set selection with random selection.

	Tiny-ImageNet	ImageNette
Core-Set	31.01	60.54
FedSD2C w/ random selection	23.32	42.06
FedSD2C	26.83	47.52

C.5 Integrating with Differential Privacy

According to [35], introducing DP-SGD [69] during the distillate synthesis stage can provide theoretical privacy guarantees for our method. We perform experiments of integrating DP-SGD in our method on Tiny-ImageNet with ResNet-18 ($\alpha = 0.1, ipc = 50$) to provide a clear view of the trade-offs involved.

Table S6: Performance of integrating DP-SGD

	$\epsilon = 1$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = \infty$
FedSD2C	22.92	25.13	26.01	26.83

D Visualization

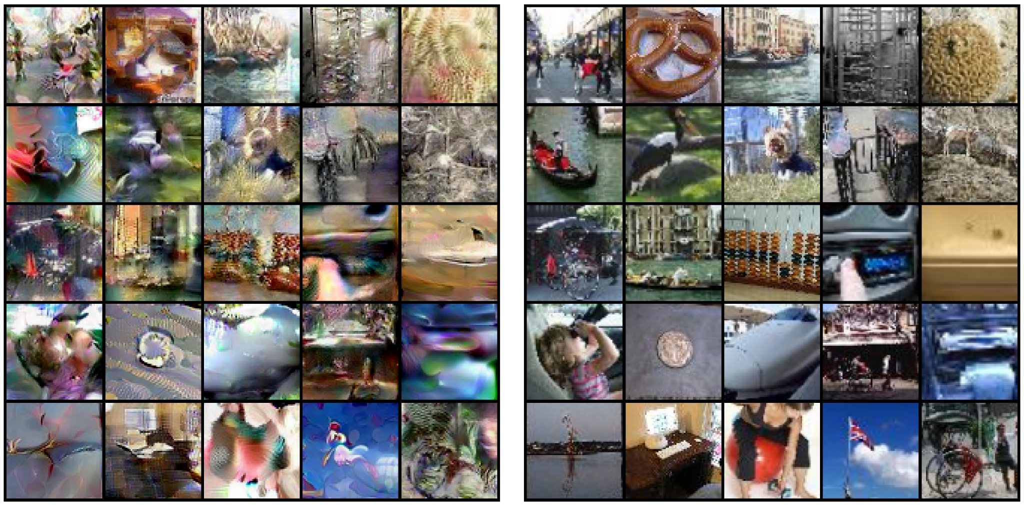


Figure S1: Visualization of synthetic distillate reconstructed by the pre-trained Autoencoder compared to the original sample on Tiny-ImageNet. The image style is similar, but with enhanced privacy protection.

https://doi.org/10.52202/079017-3259

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions are empirically validated in the main body.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Section 5

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

https://doi.org/10.52202/079017-3259

Justification: the paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide a detailed description of FedSD2C and the parameters for implementing the baseline in Section 4 and the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

https://doi.org/10.52202/079017-3259

Answer: [Yes]

Justification: Our code is attached to the supplementary material and is released at: https://github.com/Carkham/FedSD2C.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide a detailed description of FedSD2C and the parameters for implementing the baseline in Section 4 and the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the standard deviation in Table 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/079017-3259

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We describe the computational resources used in the Appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We do not violate Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential impacts about privacy issues.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

https://doi.org/10.52202/079017-3259

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all sources used and comply with their licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-3259