

# Improving Noise Efficiency in Privacy-preserving Dataset Distillation

Runkai Zheng

Carnegie Mellon University  
runkaiz@andrew.cmu.edu

Vishnu Asutosh Dasu

Pennsylvania State University  
vdasu@psu.edu

Yinong Oliver Wang

Carnegie Mellon University  
yinongwa@cs.cmu.edu

Haohan Wang

University of Illinois Urbana-Champaign  
haohanw@illinois.edu

Fernando De la Torre

Carnegie Mellon University  
ftorre@cs.cmu.edu

## Abstract

Modern machine learning models heavily rely on large datasets that often include sensitive and private information, raising serious privacy concerns. Differentially private (DP) data generation offers a solution by creating synthetic datasets that limit the leakage of private information within a predefined privacy budget; however, it requires a substantial amount of data to achieve performance comparable to models trained on the original data. To mitigate the significant expense incurred with synthetic data generation, Dataset Distillation (DD) stands out for its remarkable training and storage efficiency. This efficiency is particularly advantageous when integrated with DP mechanisms, curating compact yet informative synthetic datasets without compromising privacy. However, current state-of-the-art private DD methods suffer from a synchronized sampling-optimization process and the dependency on noisy training signals from randomly initialized networks. This results in the inefficient utilization of private information due to the addition of excessive noise. To address these issues, we introduce a novel framework that decouples sampling from optimization for better convergence and improves signal quality by mitigating the impact of DP noise through matching in an informative subspace, all without incurring additional privacy costs. On CIFAR-10, our method achieves a **10.0%** improvement with 50 images per class and **8.3%** increase with just **one-fifth** the distilled set size of previous state-of-the-art methods, demonstrating significant potential to advance privacy-preserving dataset distillation.

## 1. Introduction

In modern machine learning, large datasets are essential for training robust and accurate models. However, they

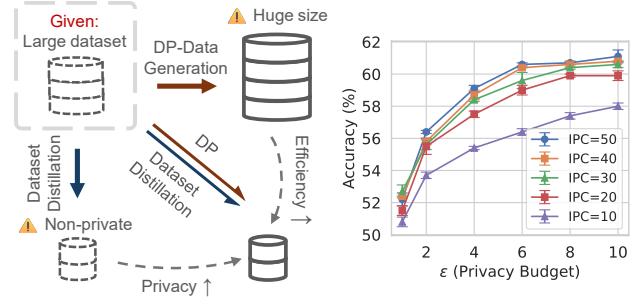


Figure 1. (a) Overview of private dataset distillation. (b) CIFAR-10 classification accuracy using distilled images with varying privacy budgets and images per class (IPC).

often contain sensitive information, posing challenges for data sharing and privacy protection. Differentially private (DP) data generation addresses this by producing synthetic datasets within a controlled privacy budget, typically using mechanisms like Differentially Private Stochastic Gradient Descent (DP-SGD) [1]. These approaches aim to approximate the data distribution and generate privacy-preserving samples. Despite notable advances, balancing utility and privacy remains difficult. Moreover, generative models often require synthesizing and storing large volumes of data to match real-data performance. For instance, Ghalebikesabi et al. [11] generates data 20x larger than the original dataset yet still underperforms it. Such volume demands lead to high storage and computational costs during training.

Dataset Distillation (DD) [31] has emerged as a promising alternative that addresses some of the inherent limitations of generative models. As fewer but highly informative synthetic samples can be retained for comparable downstream performance, DD ensures that models trained on distilled datasets perform similarly to those trained on larger original datasets with a significantly reduced storage.

While DD effectively minimizes dataset size and becomes visually anonymized, it does not inherently provide

privacy guarantees. Conversely, differentially private data generation ensures privacy but with large synthetic dataset sizes, which may not be storage or computation-efficient. This presents a critical motivation for our work: to develop a method that integrates the compactness of dataset distillation with the stringent privacy guarantees of differential privacy (Fig. 1a). Achieving this integration is challenging, as both privacy preservation and dataset compactness tend to negatively impact the utility of the dataset.

Integrating dataset distillation with differential privacy, as in methods like PSG [4] and NDPDC [39], enables private data synthesis by matching training signals (e.g., gradients, features) from the original dataset. However, these matching-based approaches face key limitations. First, they couple sampling and optimization, requiring each optimization step to be paired with new noisy queries, leading to degraded signal utility. Second, they rely on randomly initialized networks to extract training signals, which often capture uninformative details and yield low signal-to-noise ratios (SNR), amplifying DP noise effects. As a result, existing methods struggle to fully exploit limited private signals, resulting in suboptimal distilled dataset performance.

To address the limitations of matching-based DD under DP constraints, we propose a framework combining Decoupled Optimization and Sampling (DOS) with Subspace-based Error Reduction (SER) to better exploit information from private data. DOS first samples a fixed number of training signals under a DP budget, then optimizes the synthetic dataset using these precomputed signals over a separate number of iterations. Decoupling these stages allows flexible trade-offs: fewer sampling steps reduce cumulative noise, while sufficient optimization improves image quality. SER further boosts utility by projecting signals into an informative subspace learned from auxiliary data, where DP noise is injected. This concentrates signal power on high-utility dimensions, increasing the signal-to-noise ratio and mitigating DP degradation. Together, DOS and SER enhance noise efficiency, enabling compact synthetic datasets that better balance privacy and utility.

## 2. Background and Related Works

### 2.1. Differential Privacy

Differential Privacy (DP) [10] is a rigorous mathematical framework that quantifies the privacy guarantees of algorithms operating on sensitive data. A randomized mechanism  $\mathcal{M} : \mathcal{B} \rightarrow \mathcal{R}$  with domain  $\mathcal{B}$  and range  $\mathcal{R}$  satisfies  $(\epsilon, \delta)$ -differential privacy if, for any two adjacent datasets  $B, B' \in \mathcal{B}$  differing in at most one element, and for any subset of outputs  $S \subseteq \mathcal{R}$ , the following inequality holds:

$$\Pr[\mathcal{M}(B) \in S] \leq e^\epsilon \Pr[\mathcal{M}(B') \in S] + \delta.$$

where  $\epsilon \geq 0$  and  $\delta \geq 0$  are parameters that measure the strength of the privacy guarantee: the smaller they are, the

stronger the privacy. We enforce DP using the Gaussian Mechanism (GM), which perturbs a function  $f : \mathcal{B} \rightarrow \mathbb{R}^d$  by adding noise:  $\text{GM}_\sigma(B) = f(B) + \mathcal{N}(0, \sigma^2 \mathbb{I}_d)$ . Under the add/remove-one model, the  $\ell_2$ -sensitivity of  $f$  is bounded, allowing calibrated noise addition. To track cumulative privacy loss, we adopt Rényi Differential Privacy (RDP) [19] with privacy amplification via subsampling [32], and convert the results to standard  $(\epsilon, \delta)$ -DP using composition rules. Finally, we apply the post-processing property [10] to ensure that any downstream operations preserve the established privacy budget.

### 2.2. Dataset Distillation

In the context of large datasets, dataset distillation aims to reduce the dataset size while retaining the critical information needed to train a model effectively. We denote a data sample by  $\mathbf{x}$  and its label by  $\mathbf{y}$ , focusing on classification problems where  $g_\theta(\cdot)$  represents a model parameterized by  $\theta$ , and  $\ell(g_\theta(\mathbf{x}), \mathbf{y})$  denotes the cross-entropy loss between the model output  $g_\theta(\mathbf{x})$  and the label  $\mathbf{y}$ . Let  $\mathcal{D}$  and  $\mathcal{Z}$  denote the original and synthetic datasets, respectively. Our objective is to find a smaller dataset  $\mathcal{Z}$  such that training on  $\mathcal{Z}$  yields similar performance as training on  $\mathcal{D}$ . Formally, we define this dataset distillation problem as:

$$\arg \min_{\mathcal{Z}} \mathbb{E}_{(x,y) \sim \mathcal{D}} \ell(g_\theta(\mathcal{Z})(x), y),$$

where  $\theta(\mathcal{Z}) = \arg \min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{Z}} \ell(g_\theta(x), y)$ ,  $|\mathcal{Z}| \ll |\mathcal{D}|$

Following the taxonomy developed in [23], we discuss various previous methods of tackling this problem. **Meta-Model Matching** involves an inner optimization step to update model parameters  $\theta$  and an outer optimization step to refine  $\mathcal{Z}$ , aiming to make  $\mathcal{Z}$  as informative as possible for training  $\theta$ . Wang et al. [31] use stochastic gradient descent (SGD) for the inner loop and Truncated Back-propagation Through Time (TBPTT) to optimize the outer loop by unrolling a fixed number of inner loop steps. **Gradient matching methods** focus on matching the gradients of the neural network parameters when trained on synthetic data to those when trained on the original data. For instance, Zhao et al. [37] proposes optimizing synthetic data such that the gradients of a model trained on this data mimic those from the original dataset, effectively capturing essential training dynamics in a condensed form. Extensions like [36] incorporate differentiable data augmentation to enhance diversity and robustness. **trajectory matching methods** extend this idea by matching the entire training trajectory of the model parameters. Cazenavette [3] match the sequence of model states during training (the trajectory) when trained on synthetic data to those from the real data, capturing a more comprehensive view of the learning process. This approach ensures that the condensed dataset leads to similar model behavior throughout training, not

just in immediate gradients. Subsequent works build upon these concepts by integrating contrastive signals [15], aligning loss curvature [26], and scaling up to larger datasets [6], among others. **Distribution matching methods** focus on aligning feature distributions between synthetic and real datasets; [30] proposes aligning features in a latent space to improve condensation, and subsequent works minimize statistical discrepancies using metrics like Maximum Mean Discrepancy [35] or exploit attention mechanisms for efficient distillation [24]. Liu et al. [16] introduces Wasserstein distance as an alternative metric of distribution discrepancy to build a distribution matching framework. **Kernel-based distillation methods** leverage theoretical insights from kernel ridge regression and infinitely wide networks to distill datasets; foundational works like [20, 21] utilize kernel methods for condensation, while later studies improve efficiency and scalability through neural feature regression [40] and random feature approximations [17]. Other works continue to refine these approaches by incorporating implicit gradients and convex optimization techniques [9, 18]. These various methodologies reflect the diverse approaches employed in dataset distillation, each contributing unique perspectives and techniques.

### 2.3. Differentially Private Dataset Distillation

The integration of dataset distillation (DD) with differential privacy (DP) has received considerable attention in recent literature. A recent technique known as DP-KIP, proposed by Vinaroz and Park [29], utilizes DP-SGD to update synthetic data within the Kernel-Induced Points (KIP) framework, offering an effective approach for distilling private datasets. Another well-developed direction involves incorporating DP within matching-based methods, where calibrated Gaussian noise is added to the matching signal before computing matching metrics. For example, Private Set Generation (PSG) [4] introduces Gaussian noise into clipped gradients for matching, while Non-linear Differentially Private Dataset Condensation (NDPDC) [39] applies Gaussian noise to clipped features extracted from randomly initialized networks. By the post-processing theorem, these matching-based methods ensure differential privacy by aligning DP-protected signals from private datasets.

However, current matching-based DP-DD methods often couple the process of sampling signals from the private dataset with the process of optimizing the distilled images. We argue that this coupling leads to unnecessary noise addition. When sampling and optimization are performed simultaneously, their iterations are forced to be equal. When a high number of optimization iterations is required for convergence, an equally large number of sampling steps is needed. These numerous sampling steps require adding excessive noise to maintain DP. Consequently, the trade-off between iteration count and noise magnitude limits the ef-

fectiveness of these methods, as they struggle to maximize signal utility from the private dataset within a fixed privacy budget. Moreover, due to restricted access to the private training dataset, matching-based methods rely on randomly initialized neural networks to extract training signals from the private data, instead of a pre-trained network. However, randomly initialized networks capture numerous uninformative details, which lowers the signal-to-noise ratio (SNR) of the training signals. This low SNR amplifies the negative impact of added noise, further compromising the utility of the training signals and the performance of the distilled dataset.

## 3. Methodologies

Our approach maximizes the utility of training signals from two perspectives: first, we decouple the sampling process from the optimization process, allowing for extended optimization iterations without unnecessary noise addition; second, we introduce an auxiliary dataset via generative models to identify the most informative signal subspace within randomly initialized neural networks, enhancing the signal-to-noise ratio (SNR)<sup>1</sup> to reduce the impact of added noise.

### 3.1. Preliminaries and Annotations

**Private dataset** Let  $\mathcal{D} = (\mathcal{X}^{(c)}, y^{(c)})_{c=1}^C$  denote the private dataset, where:  $C$  is the total number of classes.  $\mathcal{X}^{(c)} = \{\mathbf{x}_j^{(c)}\}_{j=1}^{N^{(c)}}$  is the set of images belonging to class  $c$ .  $y^{(c)}$  is the label associated with class  $c$ .  $N^{(c)} = |\mathcal{X}^{(c)}|$  is the number of images per class (IPC).

**Synthetic Dataset** Our goal is to generate a synthetic dataset  $\mathcal{Z} = \{\mathcal{Z}^{(c)}\}_{c=1}^C$ , where  $\mathcal{Z}^{(c)} = \{\mathbf{z}_j^{(c)}\}_{j=1}^M$  represents the set of synthetic images for class  $c$ , and  $M$  is the number of synthetic IPC.

**Training Signal** We consider various types of training signals in matching-based methods, such as features in distribution-matching methods [39] and gradients in gradient-matching methods [4]. Our framework can be generalized to any type. The extraction of the signal is represented by a parameterized function  $f_\theta$ , and the final signal for the matching is denoted  $v$  and  $u$  for the real and synthetic datasets.

### 3.2. Decoupled Optimization and Sampling (DOS)

#### 3.2.1. Sampling Stage

In the sampling stage, for each class  $c$ , we perform the following steps to sample training signals at the  $i^{th}$  iteration:

1. **Data Sampling:** For each class  $c$ , sample a batch of images  $\mathcal{X}_i^{(c)} \sim \text{POISSONSAMPLE}(\mathcal{X}^{(c)}, \frac{L}{N^{(c)}})$  from the private dataset  $\mathcal{D}^{(c)}$  using Poisson sampling with probability

<sup>1</sup> Noise here in SNR refers to the uninformative features captured by randomly initialized neural networks, not the noise added for DP guarantees

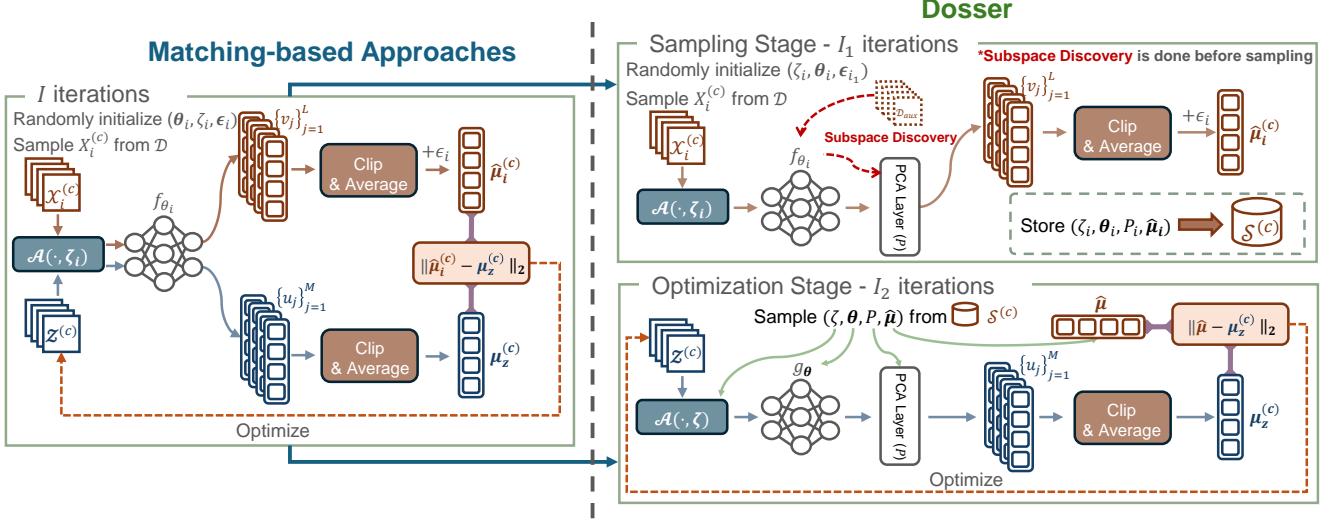


Figure 2. Overview of our proposed framework, which integrates Decoupled Optimization and Sampling (DOS) with Subspace Discovery for Error Reduction (SER).

$L/N^{(c)}$ , where  $N^{(c)} = |\mathcal{D}^{(c)}|$  is the total number of images in the dataset, and  $L$  represents the group size.

2. **Signal Extraction:** For each sampled  $j^{th}$  image  $\mathbf{x}_{i,j}^{(c)}$  at the current  $i^{th}$  iteration, apply a differentiable augmentation function  $\mathcal{A}$  using a random seed  $\zeta_i$ , denoted as  $\mathcal{A}_{\zeta_i}(\mathbf{x}_{i,j}^{(c)})$ . The augmentation function  $\mathcal{A}$  [37] includes transformations such as random cropping, color saturations, and other techniques to enhance data diversity. Next, extract training signals using a parameterized function  $f_{\theta_i}$ , where  $f_{\theta_i}$  performs feature extraction in distribution-matching methods [39] or gradient computation in gradient-matching methods [4]. The parameter set  $\theta_i$  is reinitialized for each batch sampled, allowing  $f_{\theta_i}$  to represent images across diverse signal spaces with randomly sampled parameters. To satisfy differential privacy requirements, the extracted signals are clipped to limit their sensitivity by the clipping function  $\text{clip}_K(\mathbf{v}) = \mathbf{v} \cdot \min\left(1, \frac{K}{\|\mathbf{v}\|_2}\right)$ . This ensures that the norm of  $\mathbf{v}$  does not exceed the threshold  $K$ . The entire process of extracting the training signal  $\mathbf{v}_{i,j}$  from the sampled batch  $\mathcal{X}_i^{(c)}$  can be represented by the following function:

$$\mathbf{v}_{i,j} = \mathcal{F}_{\zeta_i, \theta_i, K}(\mathbf{x}_{i,j}^{(c)}) = \text{clip}_K \circ f_{\theta_i} \circ \mathcal{A}_{\zeta_i}(\mathbf{x}_{i,j}^{(c)}). \quad (1)$$

3. **Aggregation and Noise Addition:** To ensure differential privacy, compute the aggregated signal and then add Gaussian noise:

$$\hat{\mu}_i^{(c)} = \mu_i^{(c)} + \eta_i, \quad (2)$$

where the aggregated signal  $\mu_i^{(c)}$  is defined as

$$\mu_i^{(c)} = \frac{1}{L} \sum_{\mathbf{x}_{i,j}^{(c)} \in \mathcal{X}_i^{(c)}} \mathcal{F}_{\zeta_i, \theta_i, K}(\mathbf{x}_{i,j}^{(c)}), \quad (3)$$

and  $\eta_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$  represents the Gaussian noise added for privacy. The noise scale  $\sigma$  is determined based on the desired privacy budget  $(\epsilon, \delta)$ , and the calculation of  $\sigma$  follows the process introduced by Zheng and Li [39] and through the Opacus library [34].

Following these steps, after  $I_1$  iterations, we obtain a dataset of DP-protected training signals denoted by  $\mathcal{S} = \{\mathcal{S}^{(c)}\}_{c=1}^C$  where  $C$  is the number of classes. Each subset  $\mathcal{S}^{(c)} = \{(\hat{\mu}_i, \zeta_i, \theta_i)\}_{i=1}^{I_1}$  contains  $I_1$  tuples, with each tuple consisting of:

- $\hat{\mu}_i$ : the noisy aggregated training signal,
- $\zeta_i$ : the random seed used for data augmentation,
- $\theta_i$ : the sampled model parameters at iteration  $i$ .

According to the post-processing theorem [10], any operation on the DP-protected signal set that is independent of the private data does not incur additional privacy costs. Thus, we can repeatedly use the protected signal set for subsequent optimization steps.

### 3.2.2. Optimization Stage

In the optimization stage, we utilize the stored training signals to optimize a synthetic dataset  $\mathcal{Z} = \{\mathcal{Z}^{(c)}\}_{c=1}^C$  for  $I_2$  iterations, which is initialized with random Gaussian noise. For each optimization iteration and each class  $c$ , the following steps are performed:

1. **Signal Retrieval:** Randomly select  $(\hat{\mu}_i^{(c)}, \zeta_i, \theta_i) \in \mathcal{S}^{(c)}$ .
2. **Synthetic Signal Computation:** Apply  $\mathcal{F}$  in Eq. (1) to the synthetic images with the stored random seed  $\zeta_i$ :

$$\mu_Z^{(c)} = \frac{1}{M} \sum_{\mathbf{Z}_j \in \mathcal{Z}^{(c)}} \mathcal{F}_{\zeta_i, \theta_i, K}(\mathbf{Z}_j). \quad (4)$$

3. **Loss Calculation:** Compute the squared  $\ell_2$  distance be-

tween the synthetic and noisy real aggregated signals:

$$\mathcal{L}^{(c)} = \left\| \hat{\mu}_i^{(c)} - \mu_Z^{(c)} \right\|_2^2. \quad (5)$$

4. **Parameter Update:** Update the synthetic images  $\mathcal{Z}^{(c)}$  by performing gradient descent on the loss  $\mathcal{L}^{(c)}$ .

By decoupling sampling and optimization, we can assign different numbers of iterations  $I_1$  and  $I_2$  to each process separately. This allows the optimization to converge better through longer iterations without introducing extra DP noise to the sampled signals.

### 3.3. Subspace discovery for Error Reduction (SER)

Improving the SNR in the raw extracted signal is another effective way to reduce the impact of later added DP noise. To improve the SNR, we introduce Subspace Discovery for Error Reduction (SER). SER leverages generative models to create auxiliary images that mimic the private dataset, enabling the identification of an informative subspace within a randomly initialized neural network. By projecting the signals onto the subspace, we effectively reduce the amount of noise captured by random neural networks, thereby enhancing the SNR and reducing the impact of the DP noise.

**Theoretical Insights** To understand the benefits of subspace projection in the context of differential privacy, we analyze the mean squared error (MSE) in estimating the true mean signal  $\mu$  with or without projection where the true mean of the signal  $\mu = \mathbb{E}_{x_j^{(c)} \in \mathcal{X}^{(c)}} [\mathcal{F}(x_j^{(c)})]$ . To perform this comparison, we start with the following basic assumption about the signal vector:

**Assumption 1.** Each signal vector  $v_j$ , obtained by transforming a randomly sampled real data point  $x_j$  using the function  $\mathcal{F}$ , can be modeled as:

$$v_j = \mu + p_j + r_j, \quad \|v_j\|_2 \leq K, \quad (6)$$

where  $\mu \in \mathbb{R}^D$  is the true mean signal vector,  $p_j \in \mathbb{R}^D$  represents the **informative signal** with zero mean and covariance matrix  $\Sigma_p$  (of rank  $d$ ), and  $r_j \in \mathbb{R}^D$  denotes the **uninformative signal** with zero mean and covariance  $\Sigma_r$ .

The differentially private noisy mean in the original space is calculated as  $\hat{\mu}_{\text{orig}} = \frac{1}{L} \sum_{v_i \in S} v_i + \eta_{\text{orig}}$ , where  $\eta_{\text{orig}} \sim \mathcal{N}(0, \sigma_{\text{orig}}^2 \mathbb{I}_D)$  is Gaussian noise added to satisfy a given differential privacy budget  $(\epsilon, \delta)$ . Then, the noisy mean in the projected space is obtained by  $\hat{\mu}_{\text{proj}} = \frac{1}{L} \sum_{v_i \in S} P^\top v_i + \eta_{\text{proj}}$ , where  $\eta_{\text{proj}} \sim \mathcal{N}(0, \sigma_{\text{proj}}^2 \mathbb{I}_d)$  is the Gaussian noise added in the projected space, with the same budget  $(\epsilon, \delta)$ . The noisy mean can then be reconstructed back to the original space via  $\hat{\mu}_{\text{back}} = P \hat{\mu}_{\text{proj}}$ . Under these conditions, the MSE in estimating the true mean  $\mu$  with and without projection can be defined as:

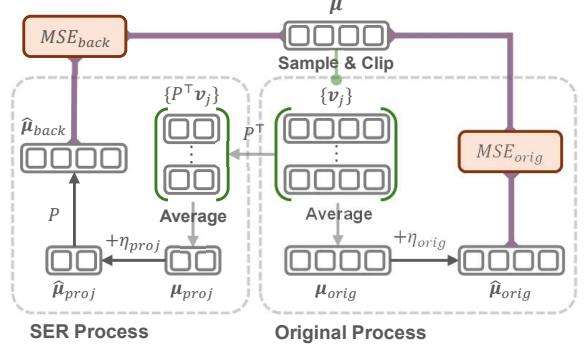


Figure 3. Illustration of Subspace discovery for Error Reduction (SER). We aim to achieve lower MSE within the differentially private framework by projecting training signals onto an informative subspace.

$$\begin{cases} \text{MSE}_{\text{orig}} = \mathbb{E} \left[ \|\hat{\mu}_{\text{orig}} - \mu\|_2^2 \right], \\ \text{MSE}_{\text{back}} = \mathbb{E} \left[ \|\hat{\mu}_{\text{back}} - \mu\|_2^2 \right]. \end{cases} \quad (7)$$

We formulate the difference between the two MSEs (as shown in Fig. 3) with the following theorem:

**Theorem 1.** Under the same budget of differential privacy  $(\epsilon, \delta)$ , the difference of MSE with and without projection  $P$  in estimating the true mean  $\mu$  can be decomposed into the three terms:

$$\begin{aligned} \text{MSE}_{\text{orig}} - \text{MSE}_{\text{back}} &= \underbrace{\frac{1}{L} \text{Tr} ((I - PP^\top) \Sigma_r)}_{\text{Projection Residual}} \\ &\quad + \underbrace{\sigma_{\text{proj}}^2 \left( \frac{\max_j \|v_j\|_2^2}{\max_j \|P^\top v_j\|_2^2} D - d \right)}_{\text{Dimensional Reduction Effect}} \\ &\quad - \underbrace{\frac{1}{L} \text{Tr} ((I - PP^\top) \mu)^2}_{\text{Projection Error}}. \end{aligned}$$

Please refer to Appendix A for the complete proof. We analyze the three terms separately as follows:

**Projection Residual:** This term captures the variance in the uninformative signal excluded by the subspace  $P$ , quantifying the uninformative components that are discarded during projection.

**Dimensional Reduction Effect:** This expression indicates that the reduction in MSE benefits from (1) the norm reduction after projection, given by  $\frac{\|v_j\|_2^2}{\|P^\top v_j\|_2^2}$ , and (2) the dimensionality reduction from  $D$  to  $d$ .

**Projection Error:** This term measures the error from projecting the true mean  $\mu$  and the informative signal variance  $\Sigma_p$  into  $P$ . It depends on the subspace dimension  $d$  and the alignment between the auxiliary and private datasets. A

well-chosen subspace minimizes this error while preserving noise reduction benefits.

In summary, the theorem shows that dimensionality reduction minimizes error in estimating the true mean by (1) discarding uninformative variance in  $\mathbf{P}$  and (2) leveraging norm and dimensionality reduction post-projection. However, projection error introduces a trade-off, potentially impacting performance due to reduced dimensions and dataset discrepancies. To address this, we propose two methods for creating an auxiliary dataset  $\mathcal{D}_{\text{aux}}$ :

**Leveraging Pre-trained Models:** We use a pre-trained foundation model such as Stable Diffusion (SD) [22] to generate images for each category. This approach is particularly effective when the target dataset's distribution closely aligns with the pre-trained model's distribution. Since it does not involve the private dataset, it also incurs no additional privacy cost. One would argue that if we already have a generative model that can produce images for a specific class, a direct approach is to generate images for distillation. The key advantage of our method is that it ensures the distilled images we generate align well with the distribution of the target dataset, capturing its unique characteristics more accurately while using images from generative models could result in distribution discrepancy.

**Using Differentially Private Generative Models:** Given a total privacy budget  $(\epsilon, \delta)$ , we allocate a portion  $(\epsilon_1, \delta_1)$  to train a generative model, such as a differentially private diffusion model (DPDM) [8], on  $\mathcal{D}$ . We then perform SER using a generated dataset by the trained model and proceed with dataset distillation under the remaining privacy budget  $(\epsilon_2, \delta_2)$ , ensuring that  $\epsilon_1 + \epsilon_2 = \epsilon$  and  $\delta_1 + \delta_2 = \delta$ . This method maintains the overall privacy budget, as formalized in the following theorem:

**Theorem 2.** *The process of distilling the private dataset  $\mathcal{D}$  with an  $(\epsilon_1, \delta_1)$ -DP mechanism, supported by SER with an auxiliary dataset  $\mathcal{D}_{\text{aux}}$  satisfying  $(\epsilon_2, \delta_2)$ -DP to  $\mathcal{D}$ , achieves  $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -DP to  $\mathcal{D}$ .*

This theorem applies the basic composition theorem (see Appendix A for the proof). Training a generative model on the private dataset requires an additional privacy cost. However, it is useful when the target domain is specialized, such as medical imaging or other niche fields not well-represented by foundational generative models typically trained on natural images.

### 3.4. Overall Framework

Our framework that combines DOS and SER for differentially private dataset distillation, named **Dosser**, is illustrated in Fig. 2. In the sampling stage for class  $c$ , for each iteration  $i$ , we initialize random neural networks and identify informative signal subspaces through PCA on the auxiliary data, giving the projection  $\mathbf{P}$ . We then sample

private data batch  $\mathcal{X}_i^{(c)}$  and obtain the training signal by  $\hat{\mu}_i^{(c)} = \frac{1}{L} \sum_j P^\top \mathcal{F}(\mathbf{x}_{i,j}^{(c)}) + \boldsymbol{\eta}_i$ . We store  $\mathbf{P}$  at each iteration  $i$  along with the data tuples, forming sets  $(\hat{\mu}_i, \zeta_i, \theta_i, \mathbf{P}_i)$  in  $\mathcal{S}$ . During the Optimization Stage, we iteratively update the synthetic dataset by aligning it with the stored noisy training signals within the identified subspaces by:

$$\mathcal{L}^{(c)} = \left\| \hat{\mu}_i^{(c)} - \mathbf{P}_i^\top \boldsymbol{\mu}_{\mathbf{Z}}^{(c)} \right\|_2^2,$$

where  $\boldsymbol{\mu}_{\mathbf{Z}}^{(c)} = \frac{1}{M} \sum_{\mathbf{z}^{(c)} \in \mathcal{Z}^{(c)}} P^\top \mathcal{F}(\mathbf{z}^{(c)})$  is the averaged signal from the synthetic dataset in the subspace. This process leverages decoupled sampling to allow extensive optimization without additional privacy costs, while subspace discovery ensures that synthetic data captures the most relevant information from the original data.

## 4. Experiments

### 4.1. Experimental Settings

**Dataset** For empirical evaluation, we use the MNIST [7], FashionMNIST [33], and CIFAR-10 [5] datasets. MNIST contains 70,000  $28 \times 28$  grayscale images of handwritten digits (0-9), with 60,000 for training and 10,000 for testing. FashionMNIST, a more challenging variant, includes 60,000  $28 \times 28$  grayscale images across 10 fashion categories, split into 50,000 training and 10,000 testing images. CIFAR-10 consists of 60,000  $32 \times 32$  color images across 10 classes, with 50,000 for training and 10,000 for testing.

**Methods** We evaluate the effectiveness of our method in comparison with several state-of-the-art differentially private data distillation approaches under a strict privacy budget of  $(\epsilon = 1, \delta = 10^{-5})$ . The methods we compare include DP-Sinkhorn [2], DP-MERF [12], PSG [4], DP-KIP-ScatterNet [29], and NDPDC [39]. As a baseline, we also compare the above-mentioned methods with standard distribution matching without differential privacy, noted as DM w/o DP, to better understand the performance gap with and without differential privacy. We implement Dosser with distribution matching, based on the framework established by Zheng and Li [39]. In our method, unless otherwise specified, we set the sampling iteration to 10,000, the optimization iteration to 200,000, and the privacy budget to  $(1, 10^{-5})$ . We also adopted Partitioning and Expansion Augmentation (PEA) from Improved Distribution Matching [38], which is a technique to enhance dataset distillation by splitting and enlarging synthetic images, to boost the performance of our framework. For SER, on MNIST and FashionMNIST, we construct an auxiliary dataset by training a differentially private diffusion model (DPDM) [8] on the private dataset with a privacy budget of  $(0.2\epsilon, 0.2\delta)$ , followed by dataset distillation with  $(0.8\epsilon, 0.8\delta)$ , ensuring an overall privacy budget of  $(\epsilon, \delta)$  as outlined in Theorem 2.

We determine the privacy-budget allocation empirically, selecting the split that yields the highest validation accuracy. The results of using other DP generators can be found in Appendix D.3. For CIFAR-10, we construct the auxiliary dataset directly using SD-v1-4 [22]. We set the subspace dimension to 500 and the auxiliary dataset size to 1000; additional details are in Appendix D.

## 4.2. Evaluation Against Baselines

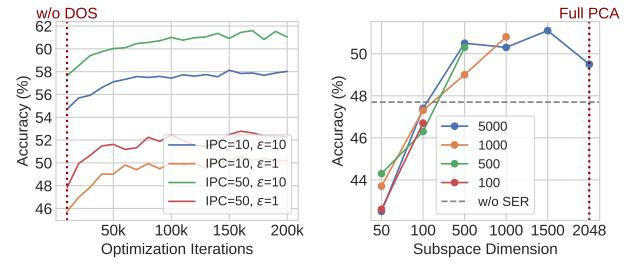
We compare the accuracies of various methods on MNIST, FashionMNIST, and CIFAR-10 under  $(1, 10^{-5})$ -DP with IPC of 10 and 50. The results are shown in Table 1. DM without differential privacy, highlighted in green, achieves the highest accuracy across datasets, showing the upper-performance limit without the added privacy constraints. Matching-based methods are highlighted in blue rows. Among them, NDPDC, which is derived from DM, exhibits noticeable accuracy degradation due to the addition of DP noise. This comparison directly highlights the impact of privacy noise on model performance. Our method, Dosser, builds upon NDPDC by enhancing the matching process with DOS and SER, improving its accuracy under the same privacy constraints. These additions increase the utility of the training signal, allowing Dosser to achieve higher accuracy than NDPDC. Specifically, Dosser provides an average improvement of 1.6% on MNIST, 2.1% on FashionMNIST, and 10.6% on CIFAR-10, with more substantial gains observed on the more complex datasets. Notably, Dosser exhibits a much smaller accuracy gap to the original DM without differential privacy. This difference is especially apparent on CIFAR-10 with IPC=10, where Dosser's performance gap from DM w/o DP is only 1.5%, compared to NDPDC's 11.7% gap. In general, Dosser's strong performance in datasets, especially with close accuracy with DM w/o DP, demonstrates its superior ability to mitigate the effects of noise within the differential privacy framework.

## 4.3. Ablation Studies

In this section, we conduct ablation studies on three key aspects: 1) evaluating the performance gain contributed by each of the proposed modules, 2) examining the effect of increasing the number of training iterations to show how DOS improves performance through additional optimization steps, and 3) analyzing the impact of varying the dimensionality of the projected subspace in SER, as well as the amount of auxiliary data used in SER.

### 4.3.1. Ablating Contributions of DOS and SER

In this study, we evaluate the individual contributions of DOS and SER to performance improvement, the results on CIFAR-10 are shown in Table 2. When applying DOS alone, accuracy improves by an average of 4.3% compared to the baseline without DOS and SER, demonstrating the advantage of additional optimization steps. Applying SER



(a) Varying optimization iterations in DOS. (b) Varying subspace dimensions and auxiliary dataset sizes in SER.

Figure 4. Ablation studies on CIFAR-10 with ConvNet.

alone also enhances results due to the increased signal-to-noise ratio introduced by SER; however, the improvement is more modest at around 1.9%. The improvement is limited when the noise required is small ( $\epsilon = 10$ ). When both DOS and SER are applied together, the combined benefits are clear: SER effectively capitalizes on the additional optimization steps provided by DOS, resulting in a 5.7% improvement over the baseline. This indicates that while SER enhances the signal-to-noise ratio, additional training iterations are essential for achieving optimal convergence. DOS and SER complement each other by enhancing the utility of training signals from two different aspects, achieving higher accuracy when combined.

### 4.3.2. Impact of DOS Hyperparameters

To investigate the impact of hyperparameters in DOS, specifically the number of optimization iterations, we analyze the accuracy changes during optimization in four typical settings varying IPC and  $\epsilon$ , while keeping  $\delta = 10^{-5}$ . The results on CIFAR-10 are presented in Figure 4a. In each setting, the sampling iteration count is set to 10,000, so the leftmost point (where the optimization iteration equals 10,000) can be regarded as the scenario without DOS. From the figure, we observe that without DOS (optimization iteration = 10,000), the optimization process of synthetic images does not fully converge, resulting in low test accuracy for downstream tasks. As the optimization iterations increase, accuracy gradually improves and ultimately reaches a much higher level than without DOS, demonstrating the necessity of decoupling sampling and optimization. The effect of DOS is particularly pronounced with higher IPC values; the accuracy gap between the initial and final evaluations is greater when IPC= 50 than when IPC= 10, regardless of whether  $\epsilon = 1$  or  $\epsilon = 10$ .

### 4.3.3. Impact of SER Hyperparameters

To examine how subspace dimensionality and the size of the auxiliary dataset in SER influence matching efficiency, we analyze downstream accuracy on CIFAR-10 across various settings, as illustrated in Figure 4b. The ConvNet feature dimension is 2048. Our results indicate minimal im-

	MNIST		FashionMNIST		CIFAR-10	
Method	IPC=10	IPC=50	IPC=10	IPC=50	IPC=10	IPC=50
DM w/o DP	97.8	99.2	84.6	88.7	52.1	60.6
DP-Sinkhorn [2]	31.7 ± 3.2	33.9 ± 1.7	9.8 ± 0.0	22.0 ± 0.1	—	—
DP-MERF [12]	75.0 ± 0.3	84.4 ± 2.3	65.5 ± 3.2	71.3 ± 1.7	—	—
DP-KIP-ScatterNet [29]	25.8 ± 2.1	13.8 ± 2.6	17.7 ± 1.5	16.2 ± 1.2	16.8 ± 1.1	9.5 ± 0.5
PSG [4]	78.6 ± 0.7	—	68.5 ± 0.5	—	33.6 ± 0.3	—
NDPDC [39]	93.1 ± 0.4	94.1 ± 0.4	77.7 ± 0.6	78.8 ± 0.4	39.4 ± 0.8	42.3 ± 0.8
<b>Dosser (ours)</b>	95.3 ± 0.0	96.4 ± 0.0	<b>81.6 ± 0.1</b>	81.8 ± 0.2	44.2 ± 0.2	49.1 ± 0.5
<b>Dosser (ours) w/ PEA [38]</b>	<b>96.4 ± 0.0</b>	<b>96.7 ± 0.1</b>	80.1 ± 0.5	<b>83.1 ± 0.5</b>	<b>50.6 ± 0.1</b>	<b>52.3 ± 0.6</b>

Table 1. The table presents a comparison of accuracies achieved by various methods on three datasets: MNIST, FashionMNIST, and CIFAR-10, evaluated under a privacy budget of  $(1, 10^{-5})$ . Each method’s performance is reported for IPC of 10 and 50.

Dosser Components		(IPC, $\epsilon$ )			
DOS	SER	(10, 1)	(50, 1)	(10, 10)	(50, 10)
✗	✗	41.7 ± 0.0	45.7 ± 0.1	54.1 ± 0.0	57.7 ± 0.0
✓	✗	47.7 ± 0.1	51.0 ± 0.2	56.7 ± 0.5	<b>61.1 ± 0.1</b>
✗	✓	46.5 ± 0.2	47.8 ± 0.3	54.7 ± 0.5	57.6 ± 0.0
✓	✓	<b>50.6 ± 0.1</b>	<b>52.3 ± 0.3</b>	<b>58.0 ± 0.2</b>	61.0 ± 0.0

Table 2. Performance improvements from individual and combined contributions of DOS and SER components under varying IPC and privacy settings on CIFAR-10.

pact from varying the auxiliary dataset size, likely because the variance in PCA parameters across different auxiliary dataset sizes is small and thus has little effect on matching performance. The only benefit of increasing dataset size is to increase the maximum dimension we can project with PCA. A notable observation is the substantial accuracy gap between using the full 2048-dimensional feature space and the results without SER. When the reduced dimensionality is set to 2048, PCA effectively performs as a single linear transformation that concentrates high-variance components in the top dimensions. This suggests that PCA’s benefits are not solely due to error reduction via dimensionality reduction but also from emphasizing high-variance components. Since the Gaussian noise is uniformly distributed across all dimensions, concentrating high-variance signals into fewer components enhances the signal-to-noise ratio in those dimensions, thereby improving matching efficiency. We conducted additional quantitative experiments to assess the direct impact of our method on noise reduction; see Appendix B for details.

## 5. Limitations

A limitation of our method is that it is specifically designed for matching training signals from randomly initialized networks, which is less competitive. Recent advances in DD, especially those scaling to larger datasets like ImageNet [14], often require pre-trained models for extracting matching signal or based on trajectory matching etc.,

which have not yet been adapted to DP-constrained scenarios. In future work, we aim to explore ways to adapt our approach to more advanced matching-based DD techniques or other state-of-the-art DD methods. Another limitation lies in SER, which requires an auxiliary dataset that closely matches the distribution of the training data. For natural image datasets or large datasets, we can create this auxiliary dataset using foundational generative models or by training a generative model with DP on the large dataset. However, for specialized domain datasets with limited data, the performance of SER may be constrained.

## 6. Conclusion

In this paper, we introduced a novel framework for differentially private dataset distillation that combines two key innovations: decoupling the sampling and optimization processes and applying subspace projection to improve signal utility. Our approach addresses the limitations of existing matching-based methods by enabling independent control over sampling and optimization iterations, which reduces cumulative noise injection and allows for more efficient utilization of the privacy budget. Additionally, our use of subspace projection identifies and focuses on the most informative signal subspace, effectively increasing the signal-to-noise ratio within each training signal. Experimental results across multiple datasets validate that our framework achieves superior accuracy and privacy efficiency compared to traditional methods. Our method offers a substantial improvement in differentially private dataset distillation, setting a new standard for privacy-preserving data synthesis.

## Acknowledgement

We would like to thank Nicholas Apostoloff, Oncel Tuzel, and Jeremy Holland for their insightful comments and constructive feedback throughout the development of this work. Their suggestions helped clarify key ideas and strengthen both the methodology and presentation. We are grateful for their time, perspective, and thoughtful engagement.

## References

- [1] 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. 1
- [2] Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don’t generate me: Training differentially private generative models with sinkhorn divergence. *Advances in Neural Information Processing Systems*, 34: 12480–12492, 2021. 6, 8
- [3] George et al. Cazenavette. Dataset distillation by matching training trajectories. In *CVPR*, 2022. 2
- [4] Dingfan Chen, Raouf Kerouche, and Mario Fritz. Private set generation with discriminative information. *Advances in Neural Information Processing Systems*, 35:14678–14690, 2022. 2, 3, 4, 6, 8
- [5] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pages 215–223. JMLR, 2011. 6
- [6] Justin et al. Cui. Scaling up dataset distillation to imagenet-1k with constant memory. In *ICML*, 2023. 3
- [7] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 6
- [8] Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *arXiv preprint arXiv:2210.09929*, 2022. 6
- [9] Jiawei Du, Yidi Jiang, Wenqing Wang, Zhenyu Qin, and Zheng-Jun Zha. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7059–7068, 2023. 3
- [10] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, 2014. 2, 4, 3
- [11] Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L Smith, Olivia Wiles, and Borja Balle. Differentially private diffusion models generate useful synthetic images. *arXiv preprint arXiv:2302.13861*, 2023. 1
- [12] Frederik Harder, Kamil Adamczewski, and Mijung Park. Dp-merf: Differentially private mean embeddings with randomfeatures for practical privacy-preserving data generation. In *International conference on artificial intelligence and statistics*, pages 1819–1827. PMLR, 2021. 6, 8
- [13] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4
- [14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25, 2012. 8
- [15] Saehyun Lee and Sung Ju Hwang. Dataset condensation with contrastive signals. In *Proceedings of the 39th International Conference on Machine Learning*, pages 12244–12254. PMLR, 2022. 3
- [16] Haoyang Liu, Yijiang Li, Tiancheng Xing, Vibhu Dalal, Luwei Li, Jingrui He, and Haohan Wang. Dataset distillation via the wasserstein metric. *arXiv preprint arXiv:2311.18531*, 2023. 3
- [17] Noel Loo, Akash Vasudevan, Alexandre Bayen, and Brandon Malone. Efficient dataset distillation using random feature approximation. In *Advances in Neural Information Processing Systems*, pages 1–13, 2022. 3
- [18] Noel Loo, Akash Vasudevan, Brandon Malone, and Alexandre Bayen. Dataset distillation with convexified implicit gradients. In *Proceedings of the 40th International Conference on Machine Learning*, pages 1–15. PMLR, 2023. 3
- [19] Ilya Mironov. Rényi differential privacy. In *IEEE CSF*, pages 263–275. IEEE, 2017. 2
- [20] Timothy Nguyen, Roman Novak, Jaehoon Lee, and Lechao Xiao. Dataset distillation with infinitely wide convolutional networks. In *Advances in Neural Information Processing Systems*, pages 5184–5197, 2021. 3
- [21] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2021. 3
- [22] 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*, pages 10684–10695, 2022. 6, 7, 4
- [23] Noveen Sachdeva and Julian McAuley. Data distillation: A survey, 2023. 2
- [24] Ahmad Sajedi, Samir Khaki, Rui Wang, et al. DataDAM: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3
- [25] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 4
- [26] Seungjae Shin, Heesun Bae, Sangwoo Kim, and Taesup Moon. Loss-curvature matching for dataset selection and condensation. In *International Conference on Artificial Intelligence and Statistics*, pages 1–14, 2023. 3
- [27] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*. 4
- [28] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 4
- [29] Margarita Vinaroz and Mijung Park. Differentially private kernel inducing points using features from scatternets (DP-KIP-scatternet) for privacy preserving data distillation. *Transactions on Machine Learning Research*, 2024. 3, 6, 8
- [30] 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*, pages 12196–12205, 2022. 3
- [31] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 2
  - [32] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019. 2
  - [33] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 6
  - [34] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021. 4
  - [35] Hansong Zhang, Shikun Li, Bo Zhao, et al. M3D: Dataset condensation by minimizing maximum mean discrepancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. 3
  - [36] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 2
  - [37] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. 2, 4
  - [38] Ganlong Zhao, Bo Zhao, and Hakan Bilen. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4903–4912, 2023. 6, 8
  - [39] Tianhang Zheng and Baochun Li. Differentially private dataset condensation. 2022. 2, 3, 4, 6, 8
  - [40] Yongchao Zhou, Jianfei Wang, Jian Chen, et al. Dataset distillation using neural feature regression. In *Advances in Neural Information Processing Systems*, pages 1–12, 2022. 3

# Improving Noise Efficiency in Privacy-preserving Dataset Distillation

## Supplementary Material

### A. Theoretical Analysis

**Lemma 1.** Let  $\mathcal{V} = \{\mathbf{v}_j\}_{j=1}^L$  be a dataset of  $L$  samples  $\mathbf{v}_i \in \mathbb{R}^D$ . Let  $\mathbf{P} \in \mathbb{R}^{D \times d}$  be a matrix with orthonormal rows (i.e.,  $\mathbf{P}\mathbf{P}^\top = \mathbb{I}_d$ ). Suppose that adding Gaussian noise  $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma_1^2 \mathbb{I}_D)$  to the sample mean ensures  $(\epsilon, \delta)$ -differential privacy:

$$\hat{\mu}_{\text{orig}} = \frac{1}{L} \sum_{j=1}^L \mathbf{v}_j + \boldsymbol{\eta}.$$

Then, adding Gaussian noise  $\boldsymbol{\eta}' \sim \mathcal{N}(0, \sigma_2^2 \mathbb{I}_d)$  to the projected sample mean:

$$\hat{\mathbf{v}}_{\text{back}} = \mathbf{P} \left( \frac{1}{L} \sum_{j=1}^L \mathbf{P}^\top \mathbf{v}_j + \boldsymbol{\eta}' \right),$$

ensures  $(\epsilon, \delta)$ -differential privacy, provided that

$$\frac{\sigma_1}{\sigma_2} = \frac{\max_j \|\mathbf{v}_j\|_2}{\max_i \|\mathbf{P}^\top \mathbf{v}_i\|_2}.$$

*Proof.* We start by recalling that the Gaussian mechanism provides  $(\epsilon, \delta)$ -differential privacy when noise drawn from  $\mathcal{N}(0, \sigma^2 \mathbb{I})$  is added to a function  $\mathcal{M}$ , where the noise scale  $\sigma$  is proportional to the function's  $\ell_2$ -sensitivity  $\Delta_{\mathcal{M}}$ .

The sensitivity of the sample mean function  $\mathcal{M}_{\text{orig}}(\mathcal{V}) = \frac{1}{L} \sum_{i=1}^L \mathbf{v}_i$  is given by

$$\Delta_{\text{orig}} = \max_{\mathcal{V}, \mathcal{V}'} \|\mathcal{M}_{\text{orig}}(\mathcal{V}) - \mathcal{M}_{\text{orig}}(\mathcal{V}')\|_2,$$

where  $\mathcal{V}$  and  $\mathcal{V}'$  differ in at most one element. The maximum change occurs when one sample is replaced, yielding

$$\Delta_{\text{orig}} = \frac{1}{L} \max_i \|\mathbf{v}_i\|_2.$$

Similarly, for the projected mean function  $\mathcal{M}_{\text{proj}}(\mathcal{V}) = \frac{1}{L} \sum_{i=1}^L \mathbf{P}^\top \mathbf{v}_i$ , the sensitivity is

$$\Delta_{\text{proj}} = \frac{1}{L} \max_i \|\mathbf{P}^\top \mathbf{v}_i\|_2.$$

The Gaussian mechanism requires the noise scale  $\sigma$  to be proportional to the sensitivity. Therefore, the ratio of the noise scales should match the ratio of sensitivities:

$$\frac{\sigma_1}{\sigma_2} = \frac{\Delta_{\text{orig}}}{\Delta_{\text{proj}}} = \frac{\max_i \|\mathbf{v}_i\|_2}{\max_i \|\mathbf{P}^\top \mathbf{v}_i\|_2}.$$

□

**Theorem 1.** Under the same budget of differential privacy  $(\epsilon, \delta)$ , the difference of MSE with and without projection  $\mathbf{P}$  in estimating the true mean  $\boldsymbol{\mu}$  can be decomposed into the three terms:

$$\begin{aligned} \text{MSE}_{\text{orig}} - \text{MSE}_{\text{back}} &= \underbrace{\frac{1}{L} \text{Tr}((I - \mathbf{P}\mathbf{P}^\top)\Sigma_r)}_{\text{Projection Residual}} \\ &\quad + \underbrace{\sigma_{\text{proj}}^2 \left( \frac{\|\mathbf{v}_j\|_2^2}{\|\mathbf{P}^\top \mathbf{v}_j\|_2^2} D - d \right)}_{\text{Dimensional Reduction Effect}} \\ &\quad - \underbrace{\|(I - \mathbf{P}\mathbf{P}^\top)\boldsymbol{\mu}\|_2^2 + \frac{1}{L} \text{Tr}((I - \mathbf{P}\mathbf{P}^\top)\Sigma_p)}_{\text{Projection Error}}. \end{aligned}$$

*Proof.* We analyze the MSE in both the original and projected spaces to establish the theorem.

First, consider the noisy mean in the original space:

$$\hat{\boldsymbol{\mu}}_{\text{orig}} = \boldsymbol{\mu} + \boldsymbol{\omega} + \boldsymbol{\eta}_{\text{orig}},$$

where  $\boldsymbol{\omega} = \frac{1}{L} \sum_j (\mathbf{p}_j + \mathbf{r}_j)$  represents the sampling deviation from the true mean due to finite sample size and inherent data variability.

The MSE in the original space is then:

$$\text{MSE}_{\text{orig}} = \mathbb{E} [\|\hat{\boldsymbol{\mu}}_{\text{orig}} - \boldsymbol{\mu}\|_2^2] = \mathbb{E} [\|\boldsymbol{\omega} + \boldsymbol{\eta}_{\text{orig}}\|_2^2].$$

Expanding the squared norm, we obtain:

$$\text{MSE}_{\text{orig}} = \mathbb{E} [\|\boldsymbol{\omega}\|_2^2] + \mathbb{E} [\|\boldsymbol{\eta}_{\text{orig}}\|_2^2] + 2\mathbb{E} [\boldsymbol{\omega}^\top \boldsymbol{\eta}_{\text{orig}}].$$

Since  $\boldsymbol{\omega}$  and  $\boldsymbol{\eta}_{\text{orig}}$  are independent and both have zero mean, the cross term vanishes:

$$\mathbb{E} [\boldsymbol{\omega}^\top \boldsymbol{\eta}_{\text{orig}}] = 0.$$

Thus, the MSE in the original space simplifies to:

$$\text{MSE}_{\text{orig}} = \mathbb{E} [\|\boldsymbol{\omega}\|_2^2] + \mathbb{E} [\|\boldsymbol{\eta}_{\text{orig}}\|_2^2].$$

Next, consider the noisy mean in the projected space:

$$\hat{\boldsymbol{\mu}}_{\text{proj}} = \mathbf{P}^\top (\boldsymbol{\mu} + \boldsymbol{\omega}) + \boldsymbol{\eta}_{\text{proj}},$$

and the reconstructed noisy mean in the original space:

$$\hat{\boldsymbol{\mu}}_{\text{back}} = \mathbf{P}\hat{\boldsymbol{\mu}}_{\text{proj}} = \mathbf{P}\mathbf{P}^\top (\boldsymbol{\mu} + \boldsymbol{\omega}) + \mathbf{P}\boldsymbol{\eta}_{\text{proj}}.$$

We introduce an error term to account for the recover error from PCA transformation. Specifically, define:

$$\xi_{\mathbf{P}} = \mathbf{P}\mathbf{P}^\top \boldsymbol{\mu} - \boldsymbol{\mu},$$

which quantifies the deviation of the true mean  $\mu$  from its projection onto the subspace spanned by  $P$ . If  $P$  perfectly captures the mean, then  $\xi_P = 0$ . Otherwise,  $\xi_P$  represents the component of  $\mu$  orthogonal to the subspace spanned by  $P$ .

Substituting this into the expression for  $\hat{\mu}_{\text{back}}$ , we obtain:

$$\hat{\mu}_{\text{back}} = \mu + PP^\top \omega + P\eta_{\text{proj}} + \xi_P.$$

The MSE in the projected and reconstructed space is therefore:

$$\begin{aligned} \text{MSE}_{\text{back}} &= \mathbb{E} [\|\hat{\mu}_{\text{back}} - \mu\|_2^2] \\ &= \mathbb{E} [\|PP^\top \omega + P\eta_{\text{proj}} + \xi_P\|_2^2] \\ &= \mathbb{E} [\|PP^\top \omega\|_2^2] + \mathbb{E} [\|P\eta_{\text{proj}}\|_2^2] + \mathbb{E} [\|\xi_P\|_2^2] \\ &\quad + 2\mathbb{E} [(PP^\top \omega)^\top (P\eta_{\text{proj}})] + 2\mathbb{E} [(PP^\top \omega)^\top \xi_P] \\ &\quad + 2\mathbb{E} [(P\eta_{\text{proj}})^\top \xi_P]. \end{aligned}$$

Given that  $\omega$ ,  $\eta_{\text{proj}}$ , and  $\xi_P$  are all zero-mean and mutually independent, the cross terms vanish:

$$\left\{ \begin{array}{l} \mathbb{E} [(PP^\top \omega)^\top (P\eta_{\text{proj}})] = 0, \\ \mathbb{E} [(PP^\top \omega)^\top \xi_P] = 0, \\ \mathbb{E} [(P\eta_{\text{proj}})^\top \xi_P] = 0. \end{array} \right.$$

Thus, the MSE in the projected and reconstructed space simplifies to:

$$\text{MSE}_{\text{back}} = \mathbb{E} [\|PP^\top \omega\|_2^2] + \mathbb{E} [\|P\eta_{\text{proj}}\|_2^2] + \mathbb{E} [\|\xi_P\|_2^2].$$

To evaluate these expectations, we consider the properties of covariance matrices. The covariance of  $\omega$  is:

$$\text{Cov}(\omega) = \frac{1}{L} (\Sigma_p + \Sigma_r).$$

Thus, the first term becomes:

$$\begin{aligned} \mathbb{E} [\|PP^\top \omega\|_2^2] &= \text{Tr}(PP^\top \text{Cov}(\omega)) \\ &= \frac{1}{L} \text{Tr}(PP^\top (\Sigma_p + \Sigma_r)). \end{aligned}$$

For the second term, since  $\eta_{\text{proj}} \sim \mathcal{N}(0, \sigma_{\text{proj}}^2 I_d)$ , we have:

$$\begin{aligned} \mathbb{E} [\|P\eta_{\text{proj}}\|_2^2] &= \text{Tr}(P^\top P \mathbb{E} [\eta_{\text{proj}} \eta_{\text{proj}}^\top]) \\ &= \text{Tr}(P^\top P \sigma_{\text{proj}}^2 I_d) \\ &= \sigma_{\text{proj}}^2 \text{Tr}(P^\top P) \\ &= \sigma_{\text{proj}}^2 d. \end{aligned}$$

The third term,  $\mathbb{E} [\|\xi_P\|_2^2]$ , quantifies the error between the mean estimated in the subspace and its projection back to the original space compared to the true mean:

$$\mathbb{E} [\|\xi_P\|_2^2] = \|\xi_P\|_2^2 = \|PP^\top \mu - \mu\|_2^2.$$

Therefore, the MSE in the projected and reconstructed space is:

$$\text{MSE}_{\text{back}} = \frac{1}{L} \text{Tr}(PP^\top (\Sigma_p + \Sigma_r)) + \sigma_{\text{proj}}^2 d + \|PP^\top \mu - \mu\|_2^2.$$

Comparing this with the MSE in the original space:

$$\text{MSE}_{\text{orig}} = \frac{1}{L} \text{Tr}(\Sigma_p + \Sigma_r) + \sigma_{\text{orig}}^2 D,$$

we define the difference  $\Delta$  as:

$$\begin{aligned} \Delta &= \text{MSE}_{\text{orig}} - \text{MSE}_{\text{back}} \\ &= \frac{1}{L} \text{Tr}(\Sigma_p + \Sigma_r) + \sigma_{\text{orig}}^2 D \\ &\quad - \left( \frac{1}{L} \text{Tr}(PP^\top (\Sigma_p + \Sigma_r)) + \sigma_{\text{proj}}^2 d + \|PP^\top \mu - \mu\|_2^2 \right). \end{aligned}$$

Simplifying the trace terms, we observe that:

$$\begin{aligned} \text{Tr}(\Sigma_p + \Sigma_r) - \text{Tr}(PP^\top (\Sigma_p + \Sigma_r)) \\ = \text{Tr}((I - PP^\top)(\Sigma_p + \Sigma_r)). \end{aligned}$$

According to Lemma 1, we have:

$$\sigma_{\text{orig}}^2 D - \sigma_{\text{proj}}^2 d = \sigma_{\text{proj}}^2 \left( \frac{\max_j \|\mathbf{v}_j\|_2^2}{\max_j \|P^\top \mathbf{v}_j\|_2^2} D - d \right).$$

Substituting above into the expression for  $\omega$ , we obtain:

$$\begin{aligned} \Delta &= \underbrace{\frac{1}{L} \text{Tr}((I - PP^\top)\Sigma_r)}_{\text{Projection Residual}} \\ &\quad + \underbrace{\sigma_{\text{proj}}^2 \left( \frac{\max_j \|\mathbf{v}_j\|_2^2}{\max_j \|P^\top \mathbf{v}_j\|_2^2} D - d \right)}_{\text{Dimensional Reduction Effect}} \\ &\quad - \underbrace{\|(I - PP^\top)\mu\|_2^2 + \frac{1}{L} \text{Tr}((I - PP^\top)\Sigma_p)}_{\text{Projection Error}}. \end{aligned}$$

□

**Theorem 2.** *thm:composition* The process of distilling the private dataset  $\mathcal{D}$  with an  $(\epsilon_1, \delta_1)$ -DP mechanism, supported by SER with an auxiliary dataset  $\mathcal{D}_{\text{aux}}$  satisfying  $(\epsilon_2, \delta_2)$ -DP to  $\mathcal{D}$ , achieves  $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -DP to  $\mathcal{D}$ .

*Proof.* To prove Theorem 2, we utilize fundamental properties of differential privacy, specifically the *Basic Composition Theorem* and the *Post-Processing Theorem*.

**Lemma 2** (Basic Composition Theorem [10]). *If a randomized mechanism  $\mathcal{M}_1$  satisfies  $(\epsilon_1, \delta_1)$ -DP and another randomized mechanism  $\mathcal{M}_2$  satisfies  $(\epsilon_2, \delta_2)$ -DP, then the sequential composition of these mechanisms, defined as  $\mathcal{M} = \mathcal{M}_2 \circ \mathcal{M}_1$ , satisfies  $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -DP.*

**Lemma 3** (Post-Processing Theorem [10]). *Any data-independent transformation of the output of a differentially private mechanism does not degrade its privacy guarantees. Formally, if  $\mathcal{M}$  satisfies  $(\epsilon, \delta)$ -DP, then for any deterministic or randomized function  $f$ , the mechanism  $f \circ \mathcal{M}$  also satisfies  $(\epsilon, \delta)$ -DP.*

We define the two mechanisms involved in the process as follows.

Let  $\mathcal{M}_1$  represent the mechanism responsible for SER. The input to  $\mathcal{M}_1$  is the private dataset  $\mathcal{D}$ , and its output is the auxiliary dataset  $\mathcal{D}_{\text{aux}}$ . By assumption,  $\mathcal{M}_1$  satisfies  $(\epsilon_1, \delta_1)$ -differential privacy with respect to  $\mathcal{D}$ .

Let  $\mathcal{M}_2$  represent the mechanism responsible for the distillation process. The inputs to  $\mathcal{M}_2$  are the private dataset  $\mathcal{D}$  and the auxiliary dataset  $\mathcal{D}_{\text{aux}}$ , and its output is the distilled dataset  $\mathcal{Z}$ . By assumption,  $\mathcal{M}_2$  satisfies  $(\epsilon_2, \delta_2)$ -differential privacy with respect to  $\mathcal{D}$ .

It is important to note that  $\mathcal{M}_2$  utilizes  $\mathcal{D}_{\text{aux}}$ , which is already the output of  $\mathcal{M}_1$ . However, since  $\mathcal{M}_1$  ensures that  $\mathcal{D}_{\text{aux}}$  is  $(\epsilon_1, \delta_1)$ -DP with respect to  $\mathcal{D}$ , any further processing of  $\mathcal{D}_{\text{aux}}$  by  $\mathcal{M}_2$  is considered post-processing of a DP-protected output.

Applying Lemma 3, the usage of  $\mathcal{D}_{\text{aux}}$  by  $\mathcal{M}_2$  does not introduce any additional privacy loss beyond what is already accounted for by  $\mathcal{M}_1$ . Therefore,  $\mathcal{M}_2$  maintains its  $(\epsilon_2, \delta_2)$ -DP guarantee with respect to  $\mathcal{D}$  independently of  $\mathcal{D}_{\text{aux}}$ .

Since  $\mathcal{M}_1$  and  $\mathcal{M}_2$  are applied sequentially, we apply Lemma 3. The cumulative privacy loss incurred by applying both mechanisms in sequence is the sum of their individual privacy parameters.

Formally, the overall mechanism  $\mathcal{M}$ , defined as:

$$\mathcal{M} = \mathcal{M}_2 \circ \mathcal{M}_1$$

satisfies:

$$\mathcal{M} \text{ satisfies } (\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)\text{-DP.}$$

By sequentially applying  $\mathcal{M}_1$  and  $\mathcal{M}_2$ , and leveraging both the Basic Composition and Post-Processing Theorems, we conclude that the combined process satisfies  $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -DP with respect to the private dataset  $\mathcal{D}$ .  $\square$

## B. Additional Quantitative Analysis

### B.1. Effect of Privacy-Budget Split

Table 3 shows how allocating the total budget ( $\epsilon=1.0$ ) between auxiliary data generation ( $\epsilon_1$ ) and DP-based optimisation ( $\epsilon_2$ ) affects downstream accuracy.

$(\epsilon_1, \epsilon_2)$	<b>(0.9, 0.1)</b>		<b>(0.8, 0.2)</b>		<b>(0.7, 0.3)</b>		<b>(0.6, 0.4)</b>		<b>(0.5, 0.5)</b>	
IPC	10	50	10	50	10	50	10	50	10	50
MNIST	96.3	96.5	96.4	96.7	95.9	96.1	94.9	95.2	93.2	94.5
FashionMNIST	80.2	82.9	80.1	83.1	79.7	82.4	78.8	80.8	76.2	79.4

Table 3. Accuracy (%) under different privacy-budget splits  $\epsilon_1+\epsilon_2=1.0$ , fixing  $\delta_1 = \delta_2 = 5 \times 10^6$ . Results show that allocating  $\epsilon_1:\epsilon_2=0.8:0.2$  offers a overall good trade-off.

### B.2. SER Performance Across Varying Noise Levels & Subspace Dimensions

Figure 5 details how the mean squared error (MSE) of mean estimation evolves on MNIST when varying both the *noise multiplier* and the number of retained *subspace dimensions* (horizontal axis in each subplot). Solid curves denote our method with SER (w/ SER); dashed curves are the vanilla DP baseline (w/o SER). A clear pattern, consistent with the residual decomposition in Theorem 1, emerges:

**Low-noise regime (noise multiplier  $\lesssim 0.4 \times 10^{-3}$ ).** Here, the DP noise injected per coordinate is small, so the total error is dominated by the *projection error* introduced by compressing and reconstructing the data. In this regime, SER can even *increase* MSE if the bottleneck is too tight; the loss of information outweighs the modest noise reduction. Consequently, retaining more subspace dimensions monotonically lowers the error, and the gap between “w/” and “w/o” SER narrows.

**High-noise regime (noise multiplier  $\gtrsim 0.9 \times 10^{-3}$ ).** When the privacy budget is tight, the additive Gaussian noise dominates. Dimensionality reduction now acts as a signal-to-noise enhancer: a lower-rank subspace filters out much of the high-dimensional noise before reconstruction. As a result, SER yields a pronounced MSE drop relative to the baseline, particularly when only a few hundred components are kept. Beyond this point, adding more dimensions simply reintroduces noise and the benefit diminishes.

**Intermediate-noise regime ( $\sim 0.5 \times 10^{-3}$  to  $0.8 \times 10^{-3}$ ).** At moderate noise levels, the two error sources balance each other. The MSE curves adopt a classic U-shape, indicative of a trade-off: MSE first decreases as noise is tamed by projection, reaches a minimum at an *optimal* dimensionality (typically 300–800 components), then increases again as projection bias begins to dominate. This turning point aligns with the crossover predicted by the dimensional-reduction effect term in Theorem 1.

Together, these three regimes offer actionable insight into how SER should be tuned in practice:

- When privacy is **loose**, favor a larger subspace or skip SER entirely.
- When privacy is **tight**, reduce dimensionality aggressively to suppress noise.
- For **intermediate** privacy budgets, select the number of

subspace dimensions that minimizes MSE.

We apply this same strategy to FashionMNIST and CIFAR-10 (Figures 6 and 7), and observe analogous trends.

## C. Qualitative Results

In Fig. 8, we present distilled samples from the CIFAR-10, FashionMNIST, and MNIST datasets. Each row corresponds to a distinct class, with all samples generated using an IPC of 10 and a privacy budget of  $(1, 10^{-5})$ .

## D. Settings for Generating Auxiliary Datasets

### D.1. Auxiliary Data Generation with Stable Diffusion (SD) [22]

For the CIFAR-10 dataset, we generate auxiliary images using Stable Diffusion version 1.4 (SD-v1-4). The generation process employs the following prompt for each category:

“A photo of a {category}”.

SD-v1-4 was trained on LAION-5B [25], a dataset that contains no information related to CIFAR-10. Therefore, using it to train CIFAR-10 is not considered a privacy leakage. Representative image samples are illustrated in Fig. 9a.

### D.2. Auxiliary Data Generation using Differentially Private Diffusion Model

For MNIST and FashionMNIST we generate auxiliary images with the Differentially Private Diffusion Model (DPDM). Concretely, we train a Noise Conditional Score Network (NCSN++)[28] for 50 epochs using Adam [13] (no weight decay), a batch size of 64, and a learning rate of  $3 \times 10^{-4}$ . The trained network is then sampled with a deterministic DDIM sampler [27] for 500 inference steps, ensuring the entire procedure conforms to the prescribed differential-privacy budget. We sample random images from the auxiliary dataset in Fig. 9b and Fig. 9c.

### D.3. Other Models as Auxiliary Data Generator

Beyond SD and DPDM, we evaluate DP-Diffusion and DP-LDM as alternative generative models for producing auxiliary datasets under differential privacy constraints. In this experiment, we generate synthetic data for MNIST, FashionMNIST, and CIFAR-10 using each model while maintaining a fixed privacy budget  $(1, 10^{-5})$ . To assess the impact of dataset size, we vary the number of images per class (IPC) between 10 and 50. The generated datasets are then used to train downstream models, following the same evaluation protocol as in previous experiments. The results are shown in Appendix D.3.

Table 4. Comparison of DP-based generative models for SER.

Dataset	DPDM		DP-Diffusion		DP-LDM	
	IPC=10	IPC=50	IPC=10	IPC=50	IPC=10	IPC=50
MNIST	<b>96.4</b>	96.7	96.3	96.7	96.3	<b>96.8</b>
FashionMNIST	80.1	83.1	80.5	83.0	<b>80.8</b>	<b>83.4</b>
CIFAR-10	47.8	<b>51.5</b>	47.5	51.0	<b>48.2</b>	51.2

## D.4. Controlling for the Impact of Extra Information in DP-Based Generative Models

To isolate the effect of additional information introduced by different generative models, we compare our method with DPDM, DP-Diffusion, and DP-LDM under the same privacy budget of  $(1, 10^{-5})$ . This comparison helps determine whether simply using these models for downstream training provides sufficient utility or if our method introduces meaningful improvements beyond what these baselines achieve. The results are shown in Appendix D.4.

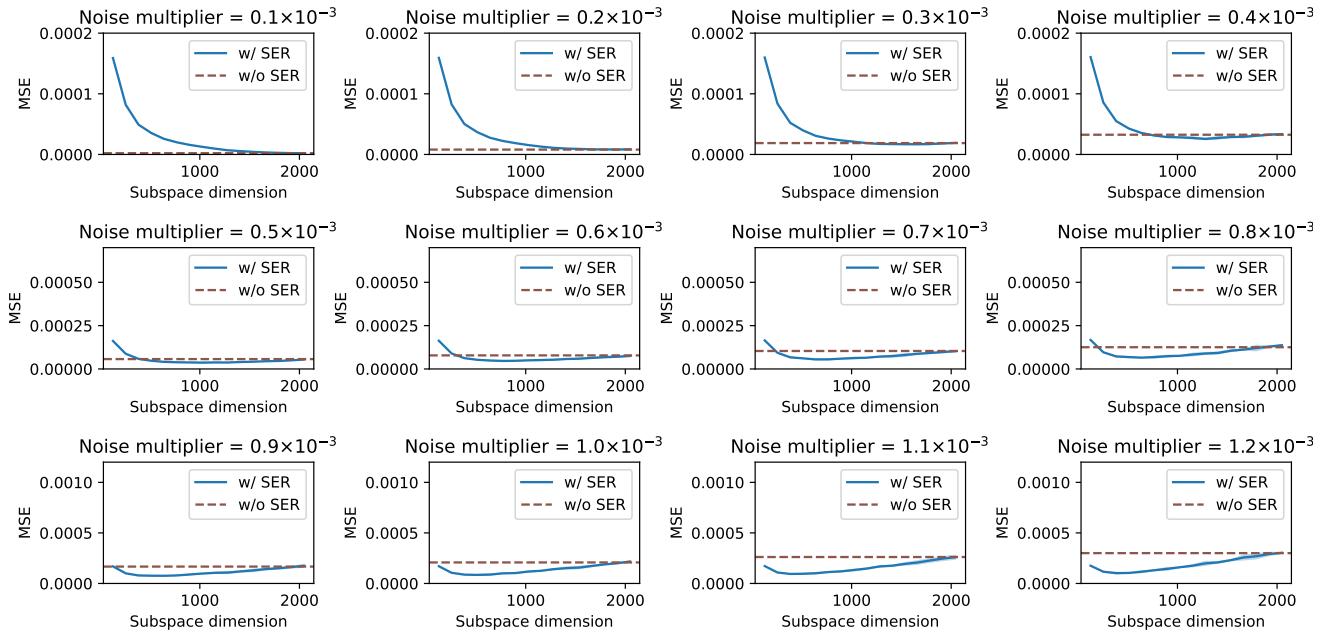
Table 5. DP-based generative models as baselines.

Dataset	Dosser		DPDM		DP-Diffusion		DP-LDM	
	IPC=10	IPC=50	IPC=10	IPC=50	IPC=10	IPC=50	IPC=10	IPC=50
MNIST	<b>96.4</b>	<b>96.7</b>	69.1	70.5	72.3	74.8	71.5	73.6
FashionMNIST	<b>80.1</b>	<b>83.1</b>	59.7	63.6	60.2	65.1	61.1	64.9
CIFAR-10	<b>50.6</b>	<b>52.3</b>	10.0	9.9	10.0	10.0	10.0	10.2

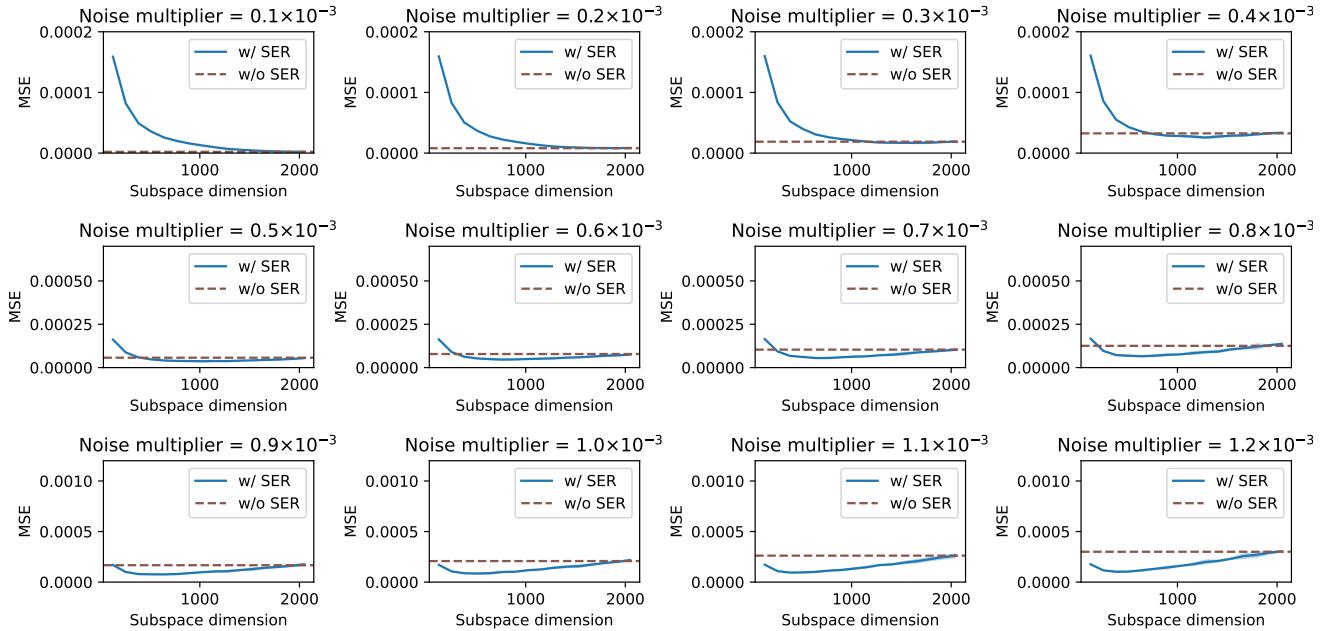
## E. Discussion

### E.1. Why Not DP-PCA for SER?

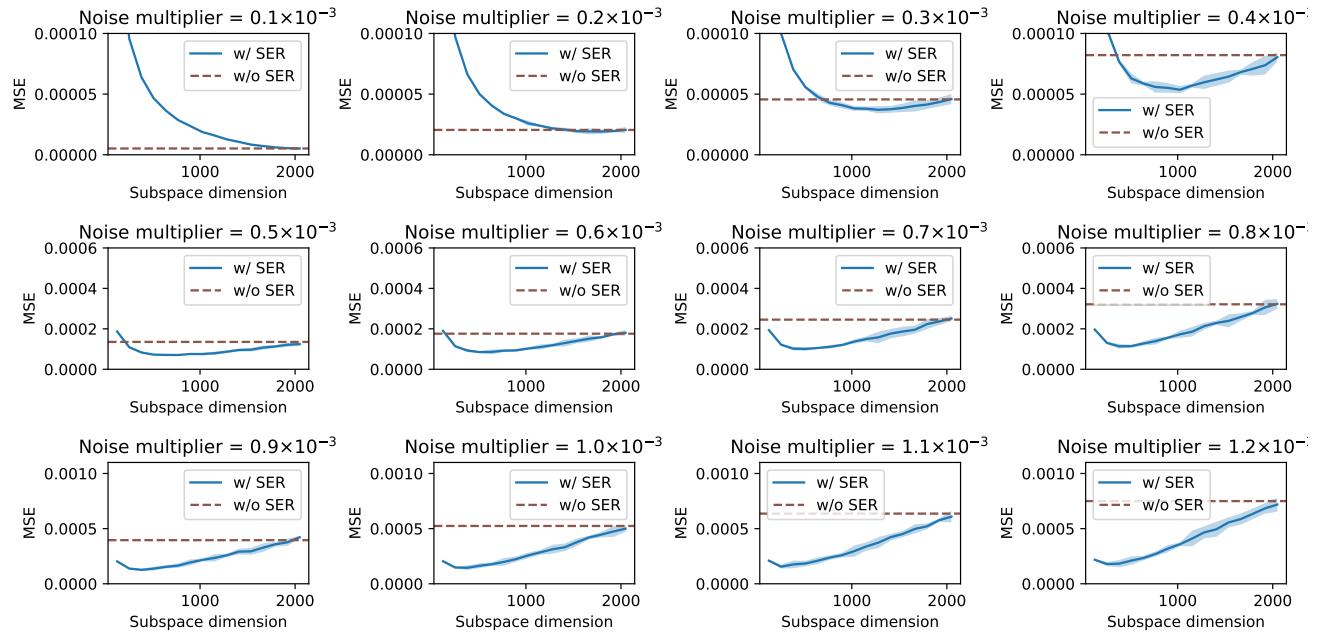
One may argue that, instead of learning a fixed projection from auxiliary data, we could simply run DP-PCA at every iteration to discover the informative subspace on-the-fly. However, because the extractor is randomly re-initialized each iteration, its output lies in a fresh feature space. Performing DP-PCA on *every* feature batch would therefore require an independent DP query each time. With a total budget  $(\epsilon, \delta)$  split across  $I$  iterations, each PCA call receives only  $\epsilon/I$  privacy, which completely drowns the signal and devastates downstream accuracy.



**Figure 5. MNIST: MSE of mean estimation as a function of retained subspace dimensions and noise multiplier.** Each subplot corresponds to a different *noise multiplier* (privacy level). The horizontal axis shows the number of retained subspace dimensions; the vertical axis shows mean-squared error (MSE). Solid curves are our method with SER; dashed curves are the vanilla DP baseline.



**Figure 6. FashionMNIST: MSE of mean estimation versus subspace dimension and noise multiplier.** Plot settings match Fig. 5. FashionMNIST exhibits the same qualitative behavior: SER offers little benefit in the low-noise regime, achieves a clear optimum in the intermediate regime (300–800 components), and substantially reduces MSE under tight privacy budgets (high noise multipliers).



**Figure 7. CIFAR-10: MSE of mean estimation versus subspace dimension and noise multiplier.** Despite the higher input dimensionality of CIFAR-10, the same trends appear: SER markedly lowers the MSE when privacy is tight (high noise), has diminishing returns as more subspace dimensions are added, and converges to the baseline when privacy is loose.

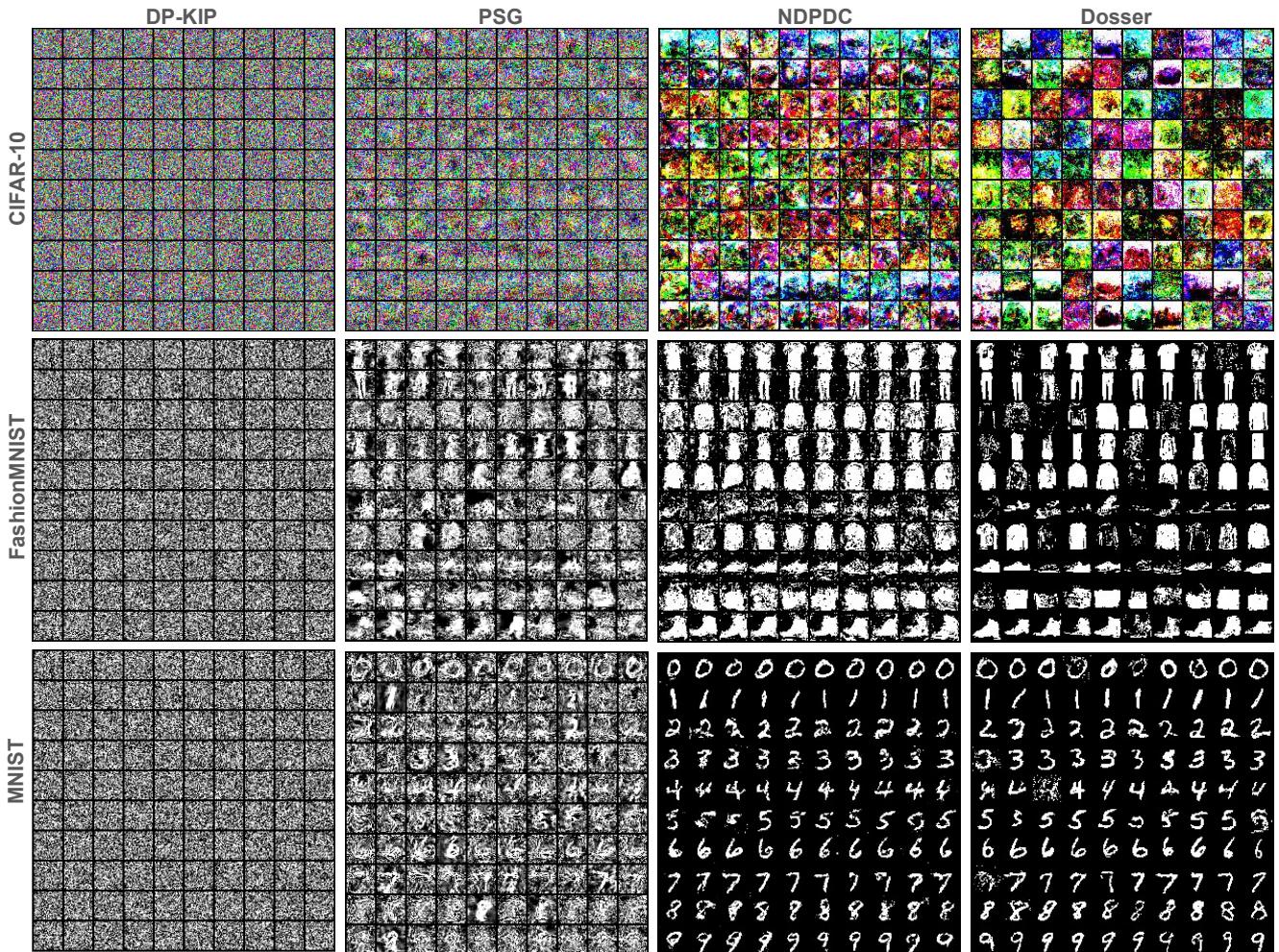
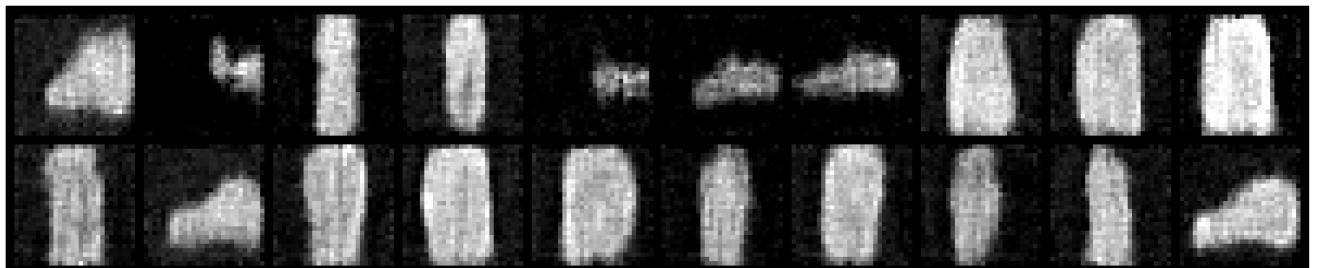


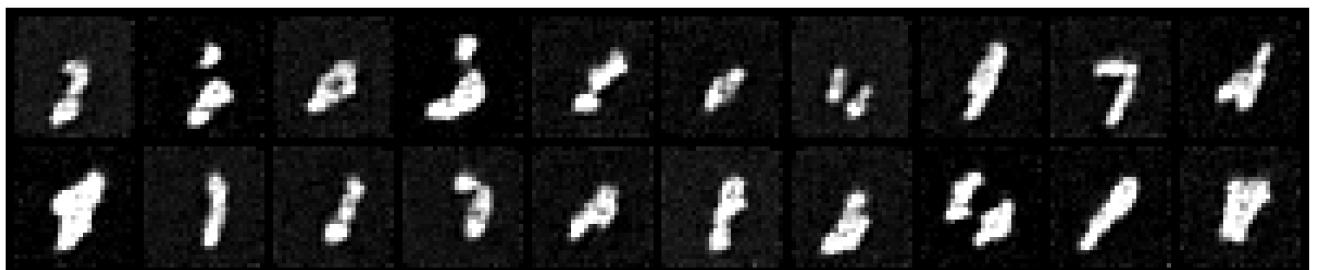
Figure 8. Distilled samples from the CIFAR-10, FashionMNIST, and MNIST datasets arranged in a  $10 \times 10$  grid. Each row represents a specific class, and all samples are generated with an IPC of 10 and a privacy budget of  $(1, 10^{-5})$ .



(a) Sampled images from the CIFAR-10 auxiliary dataset.



(b) Sampled images from the FashionMNIST auxiliary dataset.



(c) Sampled images from the MNIST auxiliary dataset.

Figure 9. Sample auxiliary datasets.