Learning Differentially Private Diffusion Models via Stochastic Adversarial Distillation

Bochao Liu[1,2], Pengju Wang[1,2], and Shiming Ge[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. While the success of deep learning relies on large amounts of training datasets, data is often limited in privacy-sensitive domains. To address this challenge, generative model learning with differential privacy has emerged as a solution to train private generative models for desensitized data generation. However, the quality of the images generated by existing methods is limited due to the complexity of modeling data distribution. We build on the success of diffusion models and introduce DP-SAD, which trains a private diffusion model by a stochastic adversarial distillation method. Specifically, we first train a diffusion model as a teacher and then train a student by distillation, in which we achieve differential privacy by adding noise to the gradients from other models to the student. For better generation quality, we introduce a discriminator to distinguish whether an image is from the teacher or the student, which forms the adversarial training. Extensive experiments and analysis clearly demonstrate the effectiveness of our proposed method.

Keywords: Generative models · Diffusion models · Differential privacy · Adversarial distillation

1 Introduction

Data sharing is essential for the development of deep learning, especially computer vision. However, in many application contexts, the sharing of data is restricted owing to its confidential nature (such as personal information on mobile devices, medical records, and financial transactions) along with strict regulatory requirements, thereby substantially impeding the advancement of technology. Data generation with differential privacy (DP) [9,10] can be a solution for data release without compromising privacy, where only a sanitized form of the data is publicly released. This sanitized synthetic data can be used as a substitute for actual data, analyzed using standard toolchains, and openly shared with the public, promoting technological progress and reproducible research in areas involving sensitive information.

Shiming Ge is the corresponding author (geshiming@iie.ac.cn).

Existing differentially private generative methods mainly focus on developing privacy-preserving generative adversarial networks (GANs), as initially introduced by [12]. They typically employ either differentially private stochastic gradient descent (DPSGD) [1], or the private aggregation of teacher ensembles (PATE) [30]. DPSGD-based methods [3,5,8,37] achieved DP by perturbing the gradients in each iteration and PATE-based methods [18,24,34] achieved DP by aggregating noise labels from teachers. These methods provided an alternative to direct data release by releasing well-trained generative models that users can use to generate data for their own downstream tasks.

However, generating high-utility data while ensuring differential privacy guarantees presents a significant challenge. There are three main shortcomings: (i) GANs are known to be considerably difficult to train, which becomes even harder when considering the privacy constraints; (ii) As the dimensionality of the data or the network escalates, an augmented quantity of noise is necessitated to attain an equivalent degree of privacy, potentially engendering more pronounced declines in performance; (iii) adding DP noise directly to all gradients introduces too much randomness, which causes damage to the quality of the generated data.

With the advent of diffusion models [16], some works [8, 11, 26] wanted to address the above shortcomings by training privacy-preserving diffusion models. Despite some achievements, it leads to new problems, where training diffusion models with differentially private algorithms (e.g. DPSGD) directly leads to excessive privacy consumption and requires pre-training on large datasets.

In this work, we abandon GANs and train a privacy-preserving diffusion model with a stochastic adversarial distillation method. As shown in Fig. 1, in contrast to existing methods, we cleverly utilize the time step of the diffusion models to dilute the effect of DP noise and combine diffusion distillation to obtain a more stable training process. Moreover, we add a discriminator to determine whether an image is generated by the teacher or the student, which can accelerate the convergence process while enhancing the quality of the data generated by the model. As an added benefit, in contrast to other DPSGD-based methods that require a large batch size to minimize the effect of DP noise, our method can take a smaller batch size and a larger time step to achieve the same effect, which allows our method to be trained in resource-constrained scenarios.

In conclusion, our DP-SAD adeptly generates privacy-preserving images by incorporating three principal components. Firstly, it employs the time step of diffusion models to reduce the impact of DP noise, while maintaining privacy without detriment to image quality. Secondly, the introduction of a discriminator facilitates adversarial training with the student model, thereby augmenting the student model's performance. Lastly, by invoking the chain rule of gradients and capitalizing on the post-processing property of differential privacy, our method effectively minimizes the introduction of randomness. These strategic implementations collectively ensure the generation of high-utility, privacy-preserving images, underscoring the efficacy and innovation of our DP-SAD.

We summarize our main contributions as follows: i) we propose a differentially private generative modeling framework named DP-SAD for effective

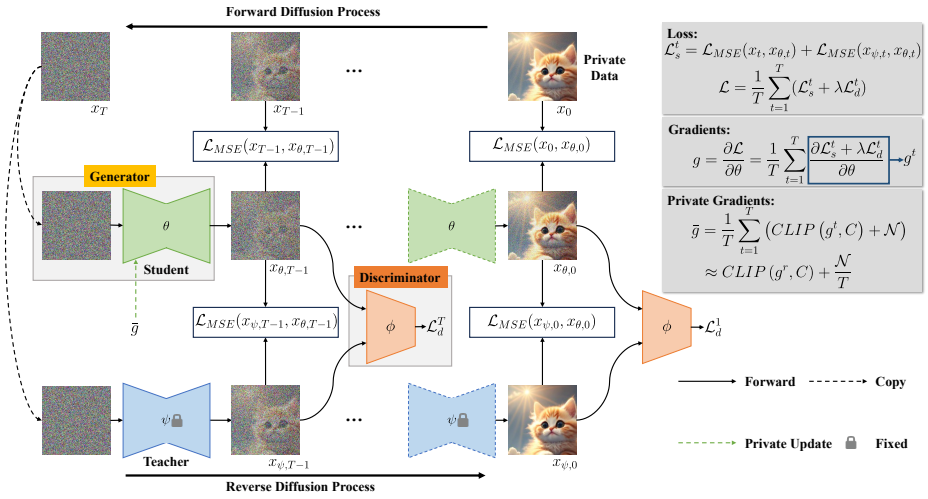

Fig. 1: Overview of our DP-SAD. We first train a teacher model ψ using the private data without protection. After that, we train a student model with the private data and the fixed teacher model in a distillation manner. In addition, we add a discriminator and view the student model as a generator to form adversarial training. Finally, for privacy, we achieve differential privacy by clipping with bound C and adding noise $\mathcal{N}$ to the gradients during backpropagation. Furthermore, we accelerate the training by using the gradient of a random time step $CLIP(g^r, C)$ instead of averaging the gradients of all time steps in the reaction process $\frac{1}{T}\sum_{t=1}^{T} CLIP(g^t, C)$, where $CLIP(*, C) = */max(1, \frac{||*||_2}{C})$.

privacy-preserving data generation; ii) we cleverly utilize the properties of the diffusion models to reduce the impact of DP noise. Combined with a discriminator, efficient and high-performance model training is achieved while allowing for resource-constrained training as an added benefit; iii) we conduct extensive experiments and analysis to demonstrate the effectiveness of our method.

2 Related Works

2.1 Diffusion Distillation.

In the advancing field of diffusion models, distillation has become a key method for enhancing model efficiency and deployment on resource-constrained platforms. Several notable works have contributed to this area by focusing on different aspects of distillation and application. Google's works [22, 31, 38] have significantly pushed the boundaries in rapid sampling and mobile device applicability. These studies highlight the potential for real-time, high-quality generative tasks on handheld devices. Further, [28,32] explored the optimization of guidance mechanisms and adversarial training in the distillation process, offering insights into the refinement of model efficiency and robustness. Innovations in data-free

distillation and quality enhancement are showcased in works like [13, 25], which proposed novel methods for minimizing dependency on large datasets and improving image resolution, respectively. Additionally, [19, 20] presented methods for reducing noise in the distillation process and enabling real-time interactive generation, highlighting the diversity of challenges and solutions in the diffusion model ecosystem. These efficient, high-quality diffusion distillation methods inspired our work, which, to our best knowledge, is the first to train a privacy-preserving diffusion model without pertaining.

2.2 Differentially Private Generative Models.

Training a DP generative model is a popular solution to the problem of privacy leakage in data sharing. Existing methods typically adopt DPSGD [3, 5, 8, 11, 26, 37] or PATE [18, 24, 34] equip the generative models with rigorous privacy guarantees. These methods, despite significant breakthroughs in the training stability problem and the visual quality problem, are far from the data utility of standardized. This is because the effect of differential privacy noise is not well minimized. Utilizing the post-processing of DP does reduce the number of additions to the noise, but does not inherently reduce its effect on the gradient. In our work, we cleverly utilize the time step of the diffusion model to dilute the effect of DP noise to improve model performance and training stability.

3 Background

3.1 Denoising Diffusion Probabilistic Models

Denoising diffusion probabilistic models [16] are recently emerged generative models that have achieved state-of-the-art results across diverse computer vision problems [2, 35]. It contains both forward and reverse processes. The forward process is a Markov chain that sequentially adds noise to a real data sample x_0 to obtain a pure noise distribution x_T, which can be understood as a labeling process. The reverse process learns the noise labels for each step in the forward process with a deep neural network to denoise x_T back to x_0.

Given a real data sample x_0, we define a posterior probability according to a variance schedule $\alpha_{[1...T]}$ as follows,

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, 1-\alpha_t\mathbf{I}), \tag{1}$$

where $\mathcal{N}(x; \mu, \sigma^2)$ represents x obeys a Gaussian distribution with μ as the mean and σ^2 as the variance. The reverse process is parameterized by a deep neural network $\epsilon_\theta(x_t, t)$ which predicts the noise ϵ added in the forward process at step t. So a simplified training loss to learn θ is as follows,

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0,t}[||\epsilon - \epsilon_\theta(x_t, t)||^2], \tag{2}$$

where $x_t = \sqrt{\prod_i \alpha_i}x_0 + \sqrt{1-\prod_i \alpha_i}\epsilon$. In inference time, model $\epsilon_\theta(\cdot)$ can denoise a pure noise distribution to a realistic image.

3.2 Differential Privacy

Differential privacy is currently an industry standard of privacy proposed by [9, 10]. It limits the extent to which the output distribution of a randomized algorithm changes in response to input changes. The following definition describes how DP provides rigorous privacy guarantees clearly.

Definition 1 (Differential Privacy). *A randomized mechanism $\mathcal{A}$ with domain $\mathcal{R}$ is (ε, δ)-differential privacy, if for all $\mathcal{O} \subseteq \mathcal{R}$ and any adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$:*

$$Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{O}] \leq e^{\varepsilon} \cdot Pr\left[\mathcal{A}\left(\mathcal{D}'\right) \in \mathcal{O}\right] + \delta, \tag{3}$$

where adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$ differ from each other with only one training example. ε is the privacy budget, which measures the degree of privacy protection of the algorithm, with smaller representing better privacy protection, and δ represents the failure probability of the algorithm, which is usually set to 10^{-5}.

Post-processing [10] is an important nature for privacy protection, which is described as follows:

Theorem 1 (Post-processing). *If mechanism $\mathcal{A}$ satisfies (ε, δ)-DP, the composition of a data-independent function $\mathcal{F}$ with $\mathcal{A}$ also satisfies (ε, δ)-DP.*

4 Method

4.1 Problem Formulation

Given a dataset $\mathcal{D} = \{x_i\}_{i=1}^n$, the objective is to train a privacy-preserving generative model ϵ_θ with parameter θ for high-utility data generation. To achieve this, we introduce a differentially private generative modeling method named DP-SAD, which contains three parts: teacher model ϵ_ψ, student model ϵ_θ and discriminator ϵ_ϕ. The training process can be formulated by minimizing an energy function $\mathbb{E}$ as follows,

$$\begin{aligned} \mathbb{E}(\epsilon_\theta; \mathcal{D}) &= \mathbb{E}_t(\epsilon_\psi; \mathcal{D}) + \mathbb{E}_s(\epsilon_\theta, \epsilon_\phi; \epsilon_\psi) \\ &= \mathbb{E}_t(\epsilon_\psi; \mathcal{D}) + \mathbb{E}_a(\epsilon_\phi; \epsilon_\psi, \epsilon_\theta) + \mathbb{E}_d(\epsilon_\theta; \epsilon_\psi, \epsilon_\phi), \end{aligned} \tag{4}$$

where teacher energy $\mathbb{E}_t$ and student energy $\mathbb{E}_s$ are used to evaluate knowledge extraction and knowledge transfer respectively. We solve it via three steps: teacher learning to achieve ϵ_ψ, adversarial learning to get ϵ_ϕ and stochastic step learning to transfer knowledge from ϵ_ψ to ϵ_θ. We emphasize that, unlike [16] where the model predicts noise added in the forward process, in this paper, models $(\epsilon_\psi, \epsilon_\theta)$ predict the image of next time step, and the two are equivalent.

4.2 Teacher Learning

We first train a teacher model using private data without any protection. This model is only used in the student training process to guide the student and is not

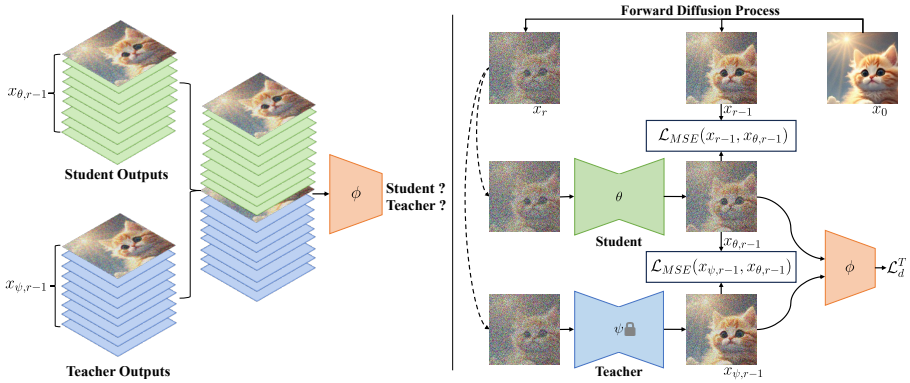

Fig. 2: Left: Illustration of the proposed discriminator. We concatenate the outputs of the teacher and student models, utilizing these combined outputs as the inputs for the discriminator. The discriminator distinguishes whether the input image originates from the teacher model or the student model. Right: Use the gradients of step r as a substitute for the average gradients over T steps. By obtaining x_r through the forward process, we prevent the teacher from inferring from noise to x_r, thereby saving computational time.

released. We follow the standard classifier-free diffusion guidance method [17] to solve the energy $\mathbb{E}_t(\epsilon_\psi; \mathcal{D})$.

$$x_t = \mathcal{S}((1+w)\epsilon_\psi(x_{t-1}, y) - w\epsilon_\psi(x_{t-1})), \tag{5}$$

where $\mathcal{S}$ is a sampler function, w is a hyperparameter and y is the label of x_{t-1}. In this way, we obtain almost the same performance as the classifier-guided diffusion model without the need for a classifier. For datasets that are either unlabeled or multi-labeled, we employ unsupervised classification methods (e.g., k-means [27]) to assign labels. In our experiments, we first utilize MoCo [7] for feature extraction, followed by the application of k-means for clustering.

4.3 Adversarial Learning

We treat the student model ϵ_θ as a generator and introduce a discriminator ϵ_ϕ to form adversarial training. The discriminator endeavors to categorize its inputs as either originating from the teacher or the student model by minimizing the following objective function [12]:

$$\mathcal{L}_{adv}^{i,t} = \log \epsilon_\phi(x_{\psi,i,t-1}) + \log(1 - \epsilon_\phi(x_{\theta,i,t-1})), \tag{6}$$

where $x_{\psi,i,t-1}$ and $x_{\theta,i,t-1}$ correspond to the $i-th$ outputs of the teacher and student models at time step $t-1$, respectively. Simultaneously, the student model aims to produce outputs closely resembling those of the teacher model, to deceive the discriminator, by minimizing the loss function $\mathcal{L}_{adv}^{i,t}$.

Since our teacher model is fixed, the first term in the loss function can be removed when updating the student model. Therefore, our loss function can be simplified as follows:

$$\mathcal{L}_{adv}^{i,t} = \log(1 - \epsilon_\phi(x_{\theta,i,t-1})). \tag{7}$$

To maintain the same format as Eq (7) when updating the discriminator, we concatenate the outputs of the teacher and student models together as the input for the discriminator as shown in the left of Fig. 2. The output corresponding to the teacher model is labeled as $[1, 0]$, while the output from the student model is labeled as $[0, 1]$. So the adversarial loss can be formulated as follows:

$$\mathcal{L}_{adv}^{i,t} = \log(1 - \epsilon_\phi(\mathcal{C}(x_{\psi,i,t-1}, x_{\theta,i,t-1}))). \tag{8}$$

where $\mathcal{C}$ represents the concatenation function. The loss function $(\mathbb{E}_a(\epsilon_\phi; \epsilon_\psi, \epsilon_\theta))$ for a batch of data over the entire time steps T is given as:

$$\mathcal{L}_{adv} = \frac{1}{B}\sum_{i=1}^{B}\left(\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{adv}^{i,t}\right) = \frac{1}{B \cdot T}\sum_{i=1}^{B}\sum_{t=1}^{T}\mathcal{L}_{adv}^{i,t}, \tag{9}$$

where B is the batch size.

4.4 Stochastic Step Distillation

As shown in Fig. 1, in addition to an adversarial loss $\mathcal{L}_{adv}$, there is also distillation loss between the outputs of the student model and the teacher model, as well as between the outputs of the student model and the original data. For this process, we adhere to the design and training method delineated in [15], a summary of which is provided herein. In time step t, we can formulate the loss as follows:

$$\mathcal{L}_{dis}^{i,t} = \mathcal{L}_{MSE}(x_{\psi,i,t-1}, x_{\theta,i,t-1}) + \mathcal{L}_{MSE}(x_{i,t-1}, x_{\theta,i,t-1}), \tag{10}$$

where $\mathcal{L}_{MSE}$ represents the mean squared error (MSE) loss. The distillation loss $(\mathbb{E}_d(\epsilon_\theta; \epsilon_\psi, \epsilon_\phi))$ for a batch of data over the entire time steps T is given as:

$$\mathcal{L}_{dis} = \frac{1}{B}\sum_{i=1}^{B}\left(\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{dis}^{i,t}\right) = \frac{1}{B \cdot T}\sum_{i=1}^{B}\sum_{t=1}^{T}\mathcal{L}_{dis}^{i,t}. \tag{11}$$

Based on the above analysis, we incorporate the adversarial loss $\mathcal{L}_{adv}$ in Eq. (9) and the distillation loss $\mathcal{L}_{dis}$ in Eq. (11) into our final loss function. Our whole framework is trained end-to-end by the following objective function:

$$\mathcal{L} = \mathcal{L}_{dis} + \lambda\mathcal{L}_{adv}, \tag{12}$$

where λ is a trade-off weight. We set it as 1 in our experiments.

Without any protection, we calculate the gradients for backpropagation as follows:

$$g = \frac{\partial \mathcal{L}}{\partial \theta} = \frac{1}{B \cdot T} \sum_{i=1}^{B} \sum_{t=1}^{T} \left(\frac{\partial (\mathcal{L}_{dis}^{i,t} + \lambda \mathcal{L}_{adv}^{i,t})}{\partial \theta} \right). \tag{13}$$

Directly updating the student model with gradients g may lead to privacy leakage. Therefore, we implement differential privacy protection by clipping it and adding noise. The specific process is as follows:

$$\begin{aligned} \bar{g} &= \frac{1}{B \cdot T} \sum_{i=1}^{B} \sum_{t=1}^{T} \left(CLIP \left(\frac{\partial (\mathcal{L}_{dis}^{i,t} + \lambda \mathcal{L}_{adv}^{i,t})}{\partial \theta}, C \right) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \right) \\ &= \frac{1}{B \cdot T} \sum_{i=1}^{B} \sum_{t=1}^{T} \left(CLIP \left(\frac{\partial (\mathcal{L}_{dis}^{i,t} + \lambda \mathcal{L}_{adv}^{i,t})}{\partial \theta}, C \right) \right) + \frac{\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})}{B \cdot T} \\ &= \frac{1}{B} \sum_{i=1}^{B} \left(\frac{1}{T} \sum_{t=1}^{T} \left(CLIP \left(\frac{\partial (\mathcal{L}_{dis}^{i,t} + \lambda \mathcal{L}_{adv}^{i,t})}{\partial \theta}, C \right) \right) \right) + \frac{\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})}{B \cdot T}, \end{aligned} \tag{14}$$

where $CLIP(*, C) = */max(1, \frac{||*||_2}{C})$. In our experiments, it was observed that an increase in the value of T correlates with an enhancement in data quality. Nonetheless, as dictated by Eq (14), each sample is subjected to T steps of diffusion throughout the training process, resulting in inefficiencies. To mitigate this, we substitute the average of the gradients over all T time steps with the gradient from a randomly selected time step.

$$\bar{g} \approx \frac{1}{B} \sum_{i=1}^{B} \left(CLIP \left(\frac{\partial (\mathcal{L}_{dis}^{i,r} + \lambda \mathcal{L}_{adv}^{i,r})}{\partial \theta}, C \right) \right) + \frac{\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})}{B \cdot T}, \tag{15}$$

where r is a number randomly selected from 0 to T. Compared to existing methods [8, 11, 26] that directly employ DPSGD to train diffusion models, our method uses the time step T to dilute the impact of noise without compromising privacy protection.

Based on previous work [5], we find that directly clipping and adding noise to each gradient introduces more randomness, leading to a decrease in the convergence speed of training. According to the properties of the chain rule for gradients, we have:

$$\frac{\partial (\mathcal{L}_{dis}^{i,r} + \lambda \mathcal{L}_{adv}^{i,r})}{\partial \theta} = \frac{\partial (\mathcal{L}_{dis}^{i,r} + \lambda \mathcal{L}_{adv}^{i,r})}{\partial x_{\theta,i,r}} \cdot \frac{\partial x_{\theta,i,r}}{\partial \theta}. \tag{16}$$

Combining the post-processing property of differential privacy, we can modify Eq (15) as follows:

$$\bar{g} \approx \frac{1}{B} \sum_{i=1}^{B} \left(CLIP \left(\frac{\partial (\mathcal{L}_{dis}^{i,r} + \lambda \mathcal{L}_{adv}^{i,r})}{\partial x_{\theta,i,r-1}}, C \right) \cdot \frac{\partial x_{\theta,i,r-1}}{\partial \theta} \right) + \frac{\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})}{B \cdot T}. \tag{17}$$

By truncating randomness in this manner, we only need to introduce randomness to $x_{\theta,i,r-1}$ once to achieve the same level of privacy protection.

4.5 Privacy Analysis

In this section, we analyze the differential privacy bound for our proposed DP-SAD and we leverage the Renyi differential privacy (RDP) [29] and Gaussian mechanism [10] in our analysis.

Definition 2 (Rényi Differential Privacy). *A randomized mechanism $\mathcal{A}$ is (q, ε)-RDP with $q > 1$ if for any adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$:*

$$D_q(\mathcal{A}(\mathcal{D})||\mathcal{A}(\mathcal{D}')) = \frac{1}{q-1}\log \mathbb{E}_{(x\sim\mathcal{A}(\mathcal{D}))}\left[\left(\frac{\Pr[\mathcal{A}(\mathcal{D})=x]}{\Pr[\mathcal{A}(\mathcal{D}')=x]}\right)^{q-1}\right] \leq \varepsilon. \tag{18}$$

Theorem 2 (Convert RDP to DP). *A (q, ε)-RDP mechanism $\mathcal{A}$ also satisfies $(\varepsilon + \log\frac{q-1}{q} - \frac{\log\delta+\log q}{q-1}, \delta)$-DP.*

Theorem 3 (Gaussian Mechanism). *Let f be a function with sensitive being $S_f = \max_{\mathcal{D},\mathcal{D}'} ||f(\mathcal{D}) - f(\mathcal{D}')||_2$ over all adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$. The Gaussian mechanism $\mathcal{A}$ with adding noise to the output of f:$\mathcal{A}(x) = f(x) + \mathcal{N}(0, \sigma^2)$ is $(q, \frac{qS_f^2}{2\sigma^2})$-RDP.*

We first calculate the sensitivity of the function that implements differential privacy. Then, based on the definitions and theories mentioned above, we derive the privacy bound of our DP-SAD.

Lemma 1. *For any neighboring gradient vectors $\bar{g}, \bar{g}'$ differing by the gradient vector of one data with length s, the l_2 sensitivity is $2C\sqrt{s}$ after performing normalization with normalization bound C.*

Proof. The l_2 sensitivity is the max change in l_2 norm caused by the input change. For the vectors after normalization with norm bound C, each dimension has a maximum value of C and a minimum value of $-C$. In the worst case, the difference of one data makes the gradient of all dimensions change from the maximum value C to the minimum value $-C$, the change in l_2 norm equals $\sqrt{(2C)^2 s} = 2C\sqrt{s}$.

We assume that the batch size is B, the number of iterations is N, and the variance of the noise added each time is σ^2.

Theorem 4. *DP-SAD guatantees $(\frac{2C^2sBN\lambda}{\sigma^2} + \log\frac{\lambda-1}{\lambda} - \frac{\log\delta+\log\lambda}{\lambda-1}, \delta)$-DP for all $\lambda \geq 1$ and $\delta \in (0, 1)$.*

Proof. For each data, the gradient clipping and noise addition implements a Gaussian mechanism which guarantees $(\lambda, \frac{2C^2s\lambda}{\sigma^2})$-RDP (Theorem 3 & Lemma 1). So the DP-SAD satisfies $(\lambda, \frac{2C^2sBN\lambda}{\sigma^2})$-RDP, which is $(\frac{2C^2sBN\lambda}{\sigma^2} + \log\frac{\lambda-1}{\lambda} - \frac{\log\delta+\log\lambda}{\lambda-1}, \delta)$-DP (Theorem 2).

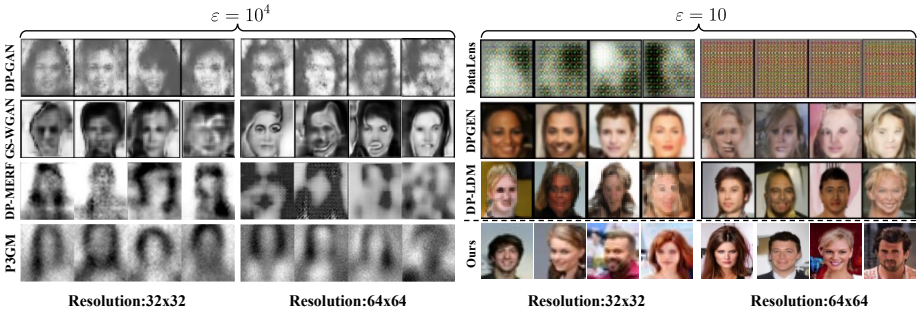

Fig. 3: Visualization results of DP-GAN, GS-WGAN, DP-MERF, P3GM, DataLens, DPGEN, DP-LDM and our DP-SAD on CelebA at 32×32 and 64×64 resolutions.

5 Experiments

To verify the effectiveness of our proposed DP-SAD, we compare it with 11 state-of-the-art methods and evaluate the data utility and visual quality on three image datasets. To ensure fair comparisons, our experiments adopt the same settings as these baselines and cite results from their original papers.

5.1 Experimental Setup

In this section, we provide a brief description of the experimental settings. For more in-depth experimental details, please refer to the supplementary material.

Datasets. We conduct experiments on three image datasets, including MNIST [21], FashionMNIST (FMNIST) [36] and CelebA [23]. To further refine our analysis, we derive two subsets from CelebA, namely CelebA-H and CelebA-G, which are created with hair color (black/blonde/brown) and gender as the label.

Baselines. We compare our DP-SAD with 11 state-of-the-art methods, including DP-GAN [37], PATE-GAN [18], DP-MERF [14], GS-WGAN [5], P3GM [33], G-PATE [24], DataLens [34], DPGEN [6], PSG [4], DP-DM [8] and DP-LDM [26].

Metrics. We evaluate our DP-SAD as well as baselines in terms of perceptual scores and classification accuracy under the same different privacy budget constraints. In particular, perceptual scores are evaluated by Inception Score (IS) and Frechet Inception Distance (FID), which are standard metrics for the visual quality of images. Classification accuracy is evaluated by training a classifier with the generated data and testing it on real test datasets

5.2 Experimental Results

Visual comparisons of generated data. We furnish visual evidence to substantiate the superior quality of data generated through our method. In Fig. 3,

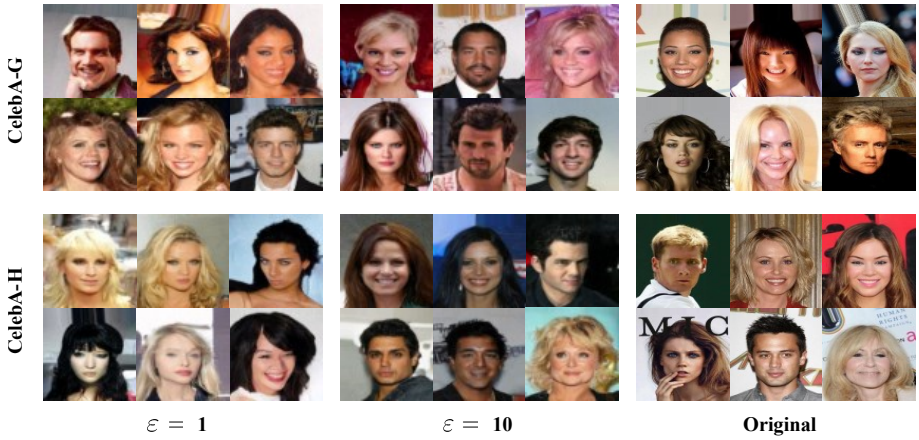

Fig. 4: Generated samples by DP-SAD on CelebA-G and CelebA-H under different privacy budget ($\varepsilon = 1$ and $\varepsilon = 10$).

we juxtapose our visualization outcomes against those derived from other benchmark models. Notably, even when operating under a stringent privacy budget condition ($\varepsilon = 10^4$), the grayscale images produced by DP-GAN, GS-WGAN, DP-MERF, and P3GM exhibit a noticeable degree of blurriness. We underscore the intrinsic advantage of grayscale images, which, due to their reduced dimensionality, facilitate a more manageable equilibrium between data quality and privacy preservation. In contrast, the color images generated by DPGEN and DP-LDM showcase a higher visual quality relative to DataLens, albeit with a lack of detailed facial features. Against this backdrop, the images emanated from our DP-SAD model distinguish themselves by presenting a more lifelike appearance coupled with enhanced facial detail, thereby validating the efficacy of our proposed method.

Image generated by DP-SAD. We present the visual quality evaluation results in Fig. 4, where all of the images were generated by DP-SAD. We find that samples at $\varepsilon = 10$ possess more facial details compared to samples at $\varepsilon = 1$. Compared to the images of 64×64 resolution presented in Fig. 3, the results of DP-SAD on images of the same resolution manifest a significantly enhanced realism and display a markedly improved facial structure. This observation highlights the ability of DP-SAD to produce more lifelike and structurally accurate facial images even under more stringent privacy settings, further evidencing the superior performance of our proposed method in generating high-quality images.

Perceptual scores comparisons. To further substantiate the efficacy of our DP-SAD, we conducted evaluations using two established metrics: IS and FID, as previously discussed. Due to the absence of experimental data for PSG and DP-DM, our comparative analysis was limited to the remaining 9 methods. The outcomes of this comparison are detailed in Tab. 1. A superior IS value is in-

Table 1: Perceptual scores comparisons with 9 state-of-the-art baselines on CelebA at 64×64 resolution under different privacy budget ε.

Method	ε	**IS**↑	**FID**↓
Without pre-training			
DP-GAN (arXiv'18)	10^4	1.00	403.94
PATE-GAN (ICLR'19)	10^4	1.00	397.62
GS-WGAN (NeurIPS'20)	10^4	1.00	384.78
DP-MERF (AISTATS'21)	10^4	1.36	327.24
P3GM (ICDE'21)	10^4	1.37	435.60
G-PATE (NeurIPS'21)	10	1.37	305.92
DataLens (CCS'21)	10	1.42	320.84
DPGEN (CVPR'22)	10	1.48	55.910
With pre-training			
DP-LDM (arXiv'23)	10	N/A	14.300
DP-SAD (Ours)	10	**2.37**	**11.260**

dicative of enhanced quality in the generated samples, whereas a diminished FID score suggests a closer resemblance to authentic images. Among the evaluated baselines, our technique distinguished itself by recording the highest IS value of 2.37 and the lowest FID score of 11.260 under the most restrictive privacy budget of 10. This can be attributed to two factors: one is that we utilized the diffusion time steps to dilute the impact of DP noise, and the other is that we incorporated a discriminator to form adversarial training. The combination of these two aspects has improved the performance of the model.

Downstream task performance comparison. Furthermore, we compare our DP-SAD with existing DP generative methods on classification tasks under two privacy budget settings $\varepsilon = 1$ and $\varepsilon = 10$ on MNIST, FMNIST, CelebA-H and CelebA-G. We evaluate the classification accuracy of the classifiers trained on the generated data, and the results are summarized in Tab. 2. It is important to note that our method does not require pre-training with any dataset. Compared to methods without pre-training, we observe consistent and significant improvements of around 4-6 percentage points across different configurations. Especially for complex tasks (CelebA) where $\varepsilon = 1$, our method outperforms other methods by at least 16 percentage points. Furthermore, when compared to the two methods with pre-training, our method consistently achieves optimal results in most settings. This improvement is attributed to the fact that we chose a more stable diffusion model instead of GANs and diluted the impact of DP noise with time steps to achieve a better balance between privacy and data utility. These results suggest that our DP-SAD can effectively generate high-quality images with practical applications.

Table 2: Classification accuracy comparisons with 11 state-of-the-art baselines under different privacy budget ε.

Method	MNIST		FMNIST		CelebA-H		CelebA-G	
	ε=1	ε=10	ε=1	ε=10	ε=1	ε=10	ε=1	ε=10
Without pre-training								
DP-GAN	0.4036	0.8011	0.1053	0.6098	0.5330	0.5211	0.3447	0.3920
PATE-GAN	0.4168	0.6667	0.4222	0.6218	0.6068	0.6535	0.3789	0.3900
GS-WGAN	0.1432	0.8075	0.1661	0.6579	0.5901	0.6136	0.4203	0.5225
DP-MERF	0.6367	0.6738	0.5862	0.6162	0.5936	0.6082	0.4413	0.4489
P3GM	0.7369	0.7981	0.7223	0.7480	0.5673	0.5884	0.4532	0.4858
G-PATE	0.5810	0.8092	0.5567	0.6934	0.6702	0.6897	0.4985	0.6217
DataLens	0.7123	0.8066	0.6478	0.7061	0.7058	0.7287	0.6061	0.6224
DPGEN	0.9046	0.9357	0.8283	0.8784	0.6999	0.8835	0.6614	0.8147
PSG	0.8090	0.9560	0.7020	0.7770	N/A	N/A	N/A	N/A
With pre-training								
DP-DM	0.9520	**0.9810**	0.7940	0.8620	N/A	N/A	N/A	N/A
DP-LDM	0.9590	0.9740	N/A	N/A	N/A	N/A	N/A	N/A
DP-SAD (Ours)	**0.9621**	0.9761	**0.8437**	**0.8960**	**0.9150**	**0.9280**	**0.8263**	**0.8414**

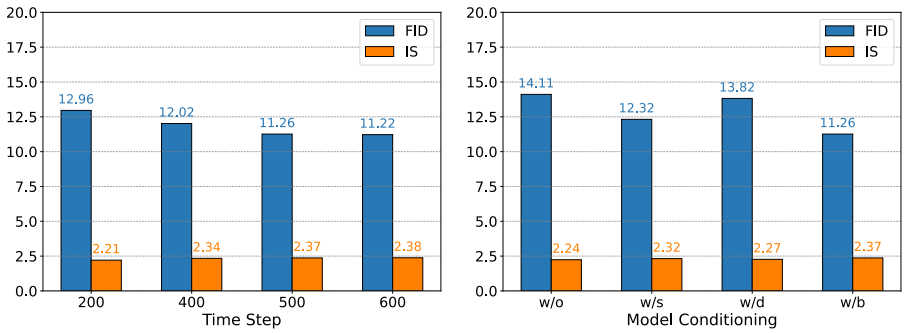

Fig. 5: Left: Perceptual scores on CelebA under different time steps. Right: Perceptual scores on CelebA under different model conditioning settings (w/o: without model conditioning, w/s: with student conditioning, w/d: with discriminator conditioning, w/b: with both model conditioning).

5.3 Ablation Studies

After the promising performance is achieved, we further analyze the impact of each component of our method, including the time step T, the model conditioning and the trade-off weight λ.

Table 3: Perceptual scores on CelebA under different trade-off weight λ.

λ	0.0	0.2	0.5	1.0	2.0	4.0
FID	14.63	13.41	12.38	11.68	12.11	13.55
IS	2.12	2.18	2.26	2.37	2.35	2.31

Impact of time step. To investigate the impact of the time step on the trade-off between privacy and data utility, we compare the perceptual scores obtained when the time step T takes different values under the same privacy budget $\varepsilon = 10$. The results are shown in Fig. 5 left. As we anticipated, with the increase in T, the IS increases and the FID decreases, which collectively indicates an improvement in image quality. This is because, on one hand, as demonstrated by Eq. (17), an increase in T will reduce the influence of noise on the gradient; on the other hand, an increase in T will enhance the generative effect of the diffusion model itself. Although increasing T improves performance, it reduces training efficiency. Therefore, in other parts of experiments, we choose $T = 500$.

Impact of model conditioning. To explore the effect of the model conditioning, including student conditioning and discriminator conditioning, we conduct experiments with/without model conditioning under the same privacy budget $\varepsilon = 10$. The results are shown in Fig. 5 right. We observe that model conditioning enhances results. Notably, student conditioning outperforms discriminator conditioning, and the combination of both student conditioning and discriminator conditioning yields the best results. An additional benefit of student conditioning is that the data inherently comes with labels when conducting downstream tasks, e.g., classifier training. Labeling data through a pre-trained model (trained with private data without any protection) may lead to privacy leakage.

Impact of λ. To study the effect of λ on the quality of the generated images, we train the student model with different trade-off weight λ under the same privacy budget $\varepsilon = 10$. The results are presented in Tab. 3. As λ increases from 0 to 1, the image quality improves with the increase in λ, as the discriminative loss $\mathcal{L}_{adv}$ drives the output distribution of the student model closer to that of the teacher model. However, when λ exceeds 1, the image quality decreases with the increase in λ. We speculate that this may be due to the larger discriminative loss $\mathcal{L}_{adv}$ constraining the efficacy of $\mathcal{L}_{dis}$. This also inspires our subsequent work, suggesting that assigning different values to λ at different time steps might be more beneficial for model training.

6 Conclusion

Direct data sharing may pose the risk of privacy leakage. To address this challenge, we proposed DP-SAD, a differentially private generative model trained by a stochastic adversarial distillation method. It achieves differential privacy

by clipping the gradients and adding noise. We ingeniously dilute the impact of noise through the diffusion model's time steps and incorporate a discriminator to form adversarial training with the student model. This method endows our model with superior performance compared to other methods. Furthermore, we combine the chain rule of gradients with the post-processing property of differential privacy to reduce the introduction of randomness, which accelerates the entire training process. Extensive experiments and analysis clearly demonstrate the effectiveness of our proposed method.

References

1. Abadi, M., Chu, A., Goodfellow, I.J., et al.: Deep learning with differential privacy. In: Proc. ACM SIGSAC Conf. Comput. Commun. Secur. pp. 308–318 (2016) 2
2. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint (2023), https://arxiv.org/abs/2311.15127 4
3. Cao, T., Bie, A., Vahdat, A., et al.: Don't generate me: Training differentially private generative models with sinkhorn divergence. In: NeurIPS. pp. 12480–12492 (2021) 2, 4
4. Chen, D., Kerkouche, R., Fritz, M.: Private set generation with discriminative information. In: NeurIPS. pp. 14678–14690 (2022) 10
5. Chen, D., Orekondy, T., Fritz, M.: GS-WGAN: A gradient-sanitized approach for learning differentially private generators. In: NeurIPS. pp. 12673–12684 (2020) 2, 4, 8, 10
6. Chen, J., Yu, C., Kao, C., et al.: DPGEN: Differentially private generative energy-guided network for natural image synthesis. In: CVPR. pp. 8387–8396 (2022) 10
7. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint (2020), https://arxiv.org/abs/2003.04297 6
8. Dockhorn, T., Cao, T., Vahdat, A., Kreis, K.: Differentially private diffusion models. arXiv preprint (2022), https://doi.org/10.48550/arXiv.2210.09929 2, 4, 8, 10
9. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: IEEE Trans. Cybern. pp. 265–284 (2006) 1, 5
10. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. pp. 211–407 (2014) 1, 5, 9
11. Ghalebikesabi, S., Berrada, L., Gowal, S., et al.: Differentially private diffusion models generate useful synthetic images. arXiv preprint (2023), https://arxiv.org/abs/2302.13861 2, 4, 8
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial nets. In: NeurIPS. pp. 2672–2680 (2014) 2, 6
13. Gu, J., Zhai, S., Zhang, Y., Liu, L., Susskind, J.M.: Boot: Data-free distillation of denoising diffusion models with bootstrapping. In: Proc. Int. Conf. Mach. Learn. Worksh. (2023) 4
14. Harder, F., Adamczewski, K., Park, M.: DP-MERF: Differentially private mean embeddings with random features for practical privacy-preserving data generation. In: International Conference on Artificial Intelligence and Statistics. pp. 1819–1827 (2021) 10

15. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NeurIPSW (2015), https://arxiv.org/abs/1503.02531 7
16. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS. pp. 6840–6851 (2020) 2, 4, 5
17. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPSW (2021) 6
18. Jordon, J., Yoon, J., Van Der Schaar, M.: PATE-GAN: Generating synthetic data with differential privacy guarantees. In: ICLR (2019), https://openreview.net/forum?id=S1zk9iRqF7 2, 4, 10
19. Katzir, O., Patashnik, O., Cohen-Or, D., Lischinski, D.: Noise-free score distillation. arXiv preprint (2023), https://arxiv.org/abs/2310.17590 4
20. Kodaira, A., Xu, C., Hazama, T., Yoshimoto, T., Ohno, K., Mitsuhori, S., Sugano, S., Cho, H., Liu, Z., Keutzer, K.: StreamDiffusion: A pipeline-level solution for real-time interactive generation. arXiv preprint (2023), https://arxiv.org/abs/2312.12491 4
21. LeCun, Y., Bottou, L., Bengio, Y., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE pp. 2278–2324 (1998) 10
22. Li, Y., Wang, H., Jin, Q., Hu, J., Chemerys, P., Fu, Y., Wang, Y., Tulyakov, S., Ren, J.: Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. In: NeurIPS (2023) 3
23. Liu, Z., Luo, P., Wang, X., et al.: Deep learning face attributes in the wild. In: ICCV. pp. 3730–3738 (2015) 10
24. Long, Y., Wang, B., Yang, Z., et al.: G-PATE: Scalable differentially private data generator via private aggregation of teacher discriminators. In: NeurIPS. pp. 2965–2977 (2021) 2, 4, 10
25. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint (2023), https://arxiv.org/abs/2310.04378 4
26. Lyu, S., Vinaroz, M., Liu, M., Park, M.: Differentially private latent diffusion models. arXiv preprint (2023), http://arxiv.org/abs/2305.15759 2, 4, 8, 10
27. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symp. Math. Stat. Prob. pp. 281–297 (1967) 6
28. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: CVPR. pp. 14297–14306 (2023) 3
29. Mironov, I.: Rényi differential privacy. In: IEEE Computer Security Foundations Symposium. pp. 263–275 (2017) 9
30. Papernot, N., Abadi, M., Erlingsson, U., et al.: Semi-supervised knowledge transfer for deep learning from private training data. In: ICLR (2017), https://arxiv.org/abs/1610.05755 2
31. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: ICLR (2022), https://arxiv.org/abs/2202.00512 3
32. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. arXiv preprint (2023), https://arxiv.org/abs/2311.17042 3
33. Takagi, S., Takahashi, T., Cao, Y., et al.: P3GM: Private high-dimensional data release via privacy preserving phased generative model. In: International Conference On Data Engineering. pp. 169–180 (2021) 10
34. Wang, B., Wu, F., Long, Y., et al.: DataLens: Scalable privacy preserving training via gradient compression and aggregation. In: Proc. ACM SIGSAC Conf. Comput. Commun. Secur. pp. 2146–2168 (2021) 2, 4, 10

35. Wang, C., Hao, Z., Tang, Y., Guo, J., Yang, Y., Han, K., Wang, Y.: SAM-DiffSR: Structure-modulated diffusion model for image super-resolution. arXiv preprint (2024), https://arxiv.org/abs/2402.17133 4
36. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv preprint (2017), https://arxiv.org/abs/1708.07747 10
37. Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J.: Differentially private generative adversarial network. arXiv preprint (2018), http://arxiv.org/abs/1802.06739 2, 4, 10
38. Zhao, Y., Xu, Y., Xiao, Z., Hou, T.: MobileDiffusion: Subsecond text-to-image generation on mobile devices. arXiv preprint (2023), https://arxiv.org/abs/2311.16567 3