dp-promise: Differentially Private Diffusion Probabilistic Models for Image Synthesis

Haichen Wang[1] *Shuchao Pang*[1*] *Zhigang Lu*[2*] *Yihang Rao*[1] *Yongbin Zhou*[1] *Minhui Xue*[3]

[1]*Nanjing University of Science and Technology, China*
[2]*James Cook University, Australia*
[3]*CSIRO's Data61, Australia*

Abstract

Utilizing sensitive images (e.g., human faces) for training DL models raises privacy concerns. One straightforward solution is to replace the private images with synthetic ones generated by deep generative models. Among all image synthesis methods, diffusion models (DMs) yield impressive performance. Unfortunately, recent studies have revealed that DMs incur privacy challenges due to the memorization of the training instances. To preserve the existence of a single private sample of DMs, many works have explored to apply DP on DMs from different perspectives. However, existing works on differentially private DMs only consider DMs as regular deep models, such that they inject unnecessary DP noise in addition to the forward process noise in DMs, damaging the model utility. To address the issue, this paper proposes Differentially Private Diffusion Probabilistic Models for Image Synthesis, dp-promise, which theoretically guarantees approximate DP by leveraging the DM noise during the forward process. Extensive experiments demonstrate that, given the same privacy budget, dp-promise outperforms the state-of-the-art on the image quality of differentially private image synthesis across the standard metrics and datasets.

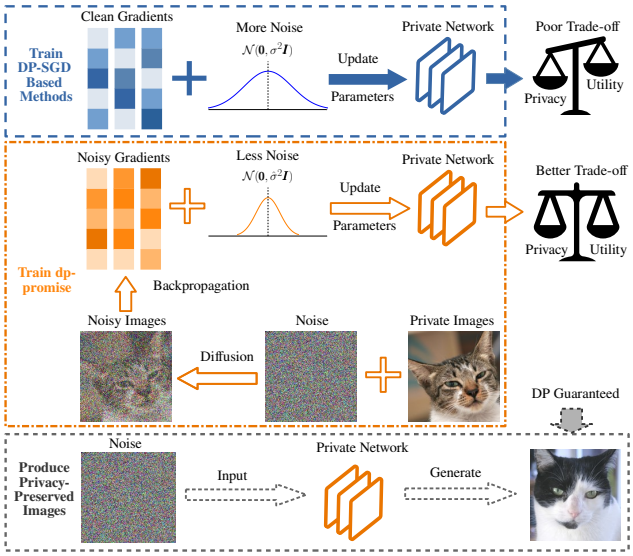


Figure 1: Comparison between dp-promise and other DP-SGD-based approaches for differentially private image synthesis using DMs.

1 Introduction

Nowadays, it is widely acknowledged that the performance of deep neural networks (DNNs) greatly benefits from large-scale training data. However, in numerous sensitive domains relying on image data, such as face recognition and medical image processing, the collection or release of a large-scale dataset often proves exceedingly challenging due to privacy issues [26, 64]. Intuitively, a straightforward solution to mitigate the risk of privacy leakage is using synthetic images, following the same distribution as the private images, to train those models. Nevertheless, recent works [6, 21] discovered that synthetic images produced by generative models, including generative adversarial networks (GANs) and diffusion models (DMs), still leak the privacy of the private data. Since DMs are more powerful than GANs in synthetic image generation, we target at preserving the privacy of training data for DMs in this paper [10].

Differential privacy [15], as a de facto privacy standard, is commonly used in preserving the individual privacy of a private dataset [41]. Specifying in machine learning tasks, differentially private stochastic gradient descent (DP-SGD) [1] incorporates gradient clipping and noise injection during the stochastic gradient descent process to ensure the privacy of training data. Later, DPDM [11] applies DP-SGD during the training procedure of DMs and then generates differentially private synthetic data. In a more recent study [17], Ghalebikesabi et al. pre-train DMs with public data and then

*Equal contribution.

fine-tune the pre-trained DMs with DP-SGD on private data. On top of DPDM, DP-LDM [37] explores the usage of latent DMs (LDMs) [44] to reduce the number of network parameters and training time while achieving similar privacy-utility trade-offs as DPDM.

Though the aforementioned studies have demonstrated the remarkable performance of DMs in differentially private image generation, they might have injected unnecessary noise when achieving DP. As shown in the upper dashed-dotted block of Figure 1, existing DPDM solutions, in general, injected both DM noise and DP noise into the DM's forward process based on the schemes of DM and DP-SGD, respectively. However, we observed that the gradients, calculated based on the DM noise-injected images, are "noisy" gradients having the same effect as DP-SGD gradients for a particular privacy budget. Hence, it is possible to achieve the same DP as the existing solutions with less DP noise injected. That is, we could inject the same amount of DP noise as the existing solutions, where partial DP noise was implemented by the DM noise. This way, we could enhance the model utility while ensuring the same DP guarantee as the existing works.

To achieve this goal, we propose a novel framework, Differentially Private Diffusion Probabilistic Models for Image Synthesis (dp-promise), which promotes the model utility by reducing the injected noise based on the theoretical analysis of the privacy guarantee provided in the DM's forward process. Specifically, dp-promise splits the normal DM forward process $[1,T]$ into two phases. In Phase I, we train DMs at $[S,T]$, where the training is conducted using a non-private optimization algorithm as usual DMs and DP noise are implemented by injecting DM noise into the images. In Phase II, we train DMs at $[1,S-1]$, where we ensure DP through DP-SGD since images at this stage do not contain enough DM noise. Furthermore, we incorporate various improvements over existing work to achieve a better trade-off between privacy and utility, e.g., freezing unnecessary layers before fine-tuning to reduce noise injection. Resulting from the above, the comprehensive experiments on four famous image datasets (MNIST, Fashion-MNIST, CelebA, and CIFAR-10 used by existing works [11, 17–19]) confirm that, given the same privacy budget, the synthetic images generated by dp-promise outperform the state-of-the-art methods roughly from 5% (compared with [17]) to 99% (compared with [19]) under commonly used performance metrics.

Our main contributions are summarized as follows.

- We propose a novel differentially private diffusion model framework, dp-promise, which provides an enhanced privacy-utility trade-off by employing a two-phase DM training process to reduce the overall noise injection.
- We provide a rigorous theoretical study on the relationship between DM noise and DP guarantees for dp-promise, making dp-promise the first work to take advantage of DM noise to achieve (approximate) DP.
- We conduct experiments on four benchmarking image datasets, including MNIST, Fashion-MNIST, CelebA, and CIFAR-10. The experimental results demonstrate that dp-promise achieves a non-trivial improvement (roughly from 5% to 99%) over state-of-the-art methods.

2 Related Work

In this section, we present a review of the related work in the field of differentially private generative models for image synthesis. We classify state-of-the-art into feature matching-based [18,19,32,52] and diffusion model-based approaches [8, 11, 17, 34, 37]. Note that there are also many works achieving DP on Generative Adversarial Networks (GANs) [7,25,33,36, 51,53,59]; however, those DP-GANs did not show promising image generation quality due to issues like mode collapse [38], hence, we do not take DP-GANs as baselines in this paper.

Approaches based on feature matching. Feature matching-based approaches utilize generative models to match the feature distributions of original data with those of generated data. Specifically, these approaches train these generative models by minimizing the Maximum Mean Discrepancy (MMD), a metric quantifying the distance between two feature distributions. To achieve DP, feature matching-based approaches inject noise into the features of the original data distribution. DP-MERF [18] trains the network by minimizing the distance of the mean embeddings between the real data and the generated data distribution, where Gaussian noise was injected into the mean embeddings of the real data distribution to achieve DP. DP-HP [52] employs hermit polynomial features instead of random Fourier features as in DP-MERF to estimate kernel mean embeddings more effectively. PEARL [32] considers training models using private embeddings constructed by a characteristic function. DP-MEPF [19] uses the perceptual feature of public data to fit a generator distribution.

Approaches based on diffusion models. Diffusion model-based approaches mainly rely on DP-SGD to achieve DP on top of diffusion models for image synthesis. DPDM [11] demonstrates how to train score-based generative models with DP-SGD and proposes a training strategy named noise multiplicity to improve performance. Both DPGEN [8] and Liu et al. [34] employ energy-based generative models trained on differentially private scores, which are constructed by randomized responses. DP-Diffusion [17] utilizes DMs pre-trained on extra public data and then fine-tunes the DMs on private data. More recently, DP-LDM [37] pre-trains latent DMs (LDMs) and fine-tunes attention modules.

Limitation. For approaches based on feature matching, e.g., DP-MERF [18] and DP-MEPF [19], they are unable to generate clear images due to noise injection on the feature of the original data distribution and then directly train the network on the noisy features to achieve DP. For approaches based

Table 1: Notation and definition.

Notation	Definition
$t \in \{1,2,\ldots,T\}$	diffusion time-steps
β_t, α_t	pre-defined diffusion noise scale
$\boldsymbol{x} \in \mathbb{R}^d$	data point in d-dimensional space
$D_{\text{priv}} = \{\boldsymbol{x}_i\}_{i=1}^n$	private dataset with n items
D_{pub}	public dataset
$\boldsymbol{z}_\theta$	neural network with parameters θ
$1 \le S \le T$	the time-step boundary between Phase I and Phase II
m_1, m_2	the batch size of Phase I and Phase II, respectively
N_1, N_2	the number of iterations in Phase I and Phase II, respectively
η_1, η_2	the learning rate in Phase I and Phase II, respectively
$\texttt{PoissionSample}_p$	Poisson sub-sampling with probability p
clip_C	clipping function with clipping constant C
σ	DP-SGD noise scale
K	the number of samples for noise augmentation
T'	the number of sampling steps
ρ	the variance hyper-parameter of sampling
w	the guidance scale of sampling

on DMs, e.g., DPDM [11] and DP-Diffusion [17] integrate DP-SGD to the training of the network in DMs but ignore the inherent privacy features within DM. Additionally, DPDM performs poorly on higher-dimensional datasets (e.g., 64×64 CelebA), and DP-Diffusion does not conduct experiments on higher-dimensional datasets.

3 Preliminaries

In this section, we present the preliminaries for this paper, including a brief introduction to DMs and the definition and properties of DP variants. Table 1 gives the definition of all the notations used in this paper.

3.1 Diffusion Models

Diffusion models (DMs) aim to learn the latent structure of a dataset by modeling the way in which data points diffuse through their latent (pixel, if in image generation task) space [50]. In a nutshell, DMs are implemented in two architectures: discrete diffusion models (denoising diffusion probabilistic models, DDPMs [22, 47]) and continuous diffusion models (score-based diffusion models [50]). In fact, the two architectures use different mathematical tools for modeling DMs, with one based on probabilistic forms and the other on stochastic differential equations (SDEs). Several works [48, 50] show that DDPM is an equivalent form of score-based DMs. Because of the high quality of image generation [40], in this paper, we use DDPM as the architecture for DMs.

Denoising diffusion probabilistic models (DDPMs) [22,47] contain a forward diffusion process and a reverse diffusion process. In the forward process, we slowly inject Gaussian noise into the original image through a series of T steps. Then, in the reverse process, we aim to learn models that reconstruct the real/original image from noisy images produced in the forward process. Formally, given an original data point $\boldsymbol{x}^{(0)}$, the forward process can be represented as a Markov chain $\{\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}\}$, where $\boldsymbol{x}^{(t)}$, at diffusion time-step $t \in \{1,2,\ldots,T\}$, was produced by injecting carefully scaled Gaussian noise into $\boldsymbol{x}^{(t-1)}$ using noise scale $\{\beta_t\}$. For the entire forward process, the posterior $q(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(T)} | \boldsymbol{x}^{(0)})$ holds that

$$q(\boldsymbol{x}^{(t)} | \boldsymbol{x}^{(t-1)}) = \mathcal{N}(\boldsymbol{x}^{(t)}; \sqrt{1-\beta_t}\boldsymbol{x}^{(t-1)}, \beta_t \boldsymbol{I}), \tag{1}$$

$$q(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(T)} | \boldsymbol{x}^{(0)}) = \prod_{t=1}^{T} q(\boldsymbol{x}^{(t)} | \boldsymbol{x}^{(t-1)}). \tag{2}$$

It is worth noting that, as the time-step t approaches T, the noisy data $\boldsymbol{x}^{(t)}$ approximates a standard normal distribution due to the cumulative impact of the injected noise. By utilizing the inherent properties of the Gaussian distribution, we have

$$q(\boldsymbol{x}^{(t)} | \boldsymbol{x}^{(0)}) = \mathcal{N}(\boldsymbol{x}^{(t)}; \sqrt{\alpha_t}\boldsymbol{x}^{(0)}, (1-\alpha_t)\boldsymbol{I}), \tag{3}$$

where $\alpha_t = \prod_{s=1}^{t}(1-\beta_s)$. Following Equation (3), the noisy data at time-step t can be calculated as

$$\boldsymbol{x}^{(t)} = \sqrt{\alpha_t}\boldsymbol{x}^{(0)} + \sqrt{1-\alpha_t}\boldsymbol{z}, \tag{4}$$

where random noise $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

In DMs, the reverse process is also defined as a Markov chain $\{\boldsymbol{x}^{(T)}, \boldsymbol{x}^{(T-1)}, \ldots, \boldsymbol{x}^{(0)}\}$. We leverage Denoising Diffusion Implicit Models (DDIMs) [48] as the reverse process to predict the noise distribution between two adjacent time-steps t and $t-1$, then recover $\boldsymbol{x}^{(t-1)}$ from $\boldsymbol{x}^{(t)}$ by removing the noise. To do so, we first learn models $\boldsymbol{z}_\theta$ to predict noise injected between $\boldsymbol{x}^{(0)}$ and $\boldsymbol{x}^{(t)}$, $t \in \{1,2,\ldots,T\}$, then derive the mean of the predicted distribution as

$$\hat{\boldsymbol{\mu}}_\theta(\boldsymbol{x}^{(t)}, t) = \sqrt{\alpha_{t-1}}\tilde{\boldsymbol{x}}^{(0)} + \sqrt{1-\alpha_{t-1}-\hat{\Sigma}_t^2} \cdot \boldsymbol{z}_\theta(\boldsymbol{x}^{(t)}, t), \tag{5}$$

where $\hat{\Sigma}_t = \rho\sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{\beta_t}$, $\tilde{\boldsymbol{x}}^{(0)} = (\boldsymbol{x}^{(t)} - \sqrt{1-\alpha_t}\boldsymbol{z}_\theta(\boldsymbol{x}^{(t)}, t))/\sqrt{\alpha_t}$, and the hyper-parameter ρ controls variance during sampling procedure. After having $\hat{\boldsymbol{\mu}}_\theta(\boldsymbol{x}^{(t)}, t)$, we will predict the noise distribution immediately as $\hat{p}_\theta(\boldsymbol{x}^{(t-1)} | \boldsymbol{x}^{(t)}) = \mathcal{N}(\boldsymbol{x}^{(t-1)}; \hat{\boldsymbol{\mu}}_\theta(\boldsymbol{x}^{(t)}, t), \hat{\Sigma}_t^2 \boldsymbol{I})$.

To train the neural network, the parameters θ are optimized by the following objective

$$\arg\min_{\theta} \mathbb{E}_{t,\boldsymbol{x}^{(0)},\boldsymbol{z}}\left[\|\boldsymbol{z}-\boldsymbol{z}_\theta(\sqrt{\alpha_t}\boldsymbol{x}^{(0)}+\sqrt{1-\alpha_t}\boldsymbol{z},t)\|_2^2\right], \quad (6)$$

where t is uniformly selected from $\{1,2,\ldots,T\}$, $\boldsymbol{x}^{(0)} \sim D$, and $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$. In practice, to reconstruct the desired data from noise, denoted as $\hat{\boldsymbol{x}}^{(T)} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$, we iteratively generate the synthetic data as follows

$$\begin{aligned}\hat{\boldsymbol{x}}^{(t-1)} = \sqrt{\alpha_{t-1}}\frac{\hat{\boldsymbol{x}}^{(t)}-\sqrt{1-\alpha_t}\boldsymbol{z}_\theta(\hat{\boldsymbol{x}}^{(t)},t)}{\sqrt{\alpha_t}}+ \\ \sqrt{1-\alpha_{t-1}-\hat{\Sigma}_t^2}\cdot\boldsymbol{z}_\theta(\hat{\boldsymbol{x}}^{(t)},t)+\hat{\Sigma}_t\boldsymbol{\xi},\end{aligned} \quad (7)$$

where t ranges from T to 1, and $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$. Note that in DDIM, the synthetic data generation process can be accelerated by sampling in a small subset of time-steps $\{\tau_1,\tau_2,\ldots,\tau_{T'}\}$ instead of the entire diffusion time-steps as in DDPM.

3.2 Differential Privacy

Differential privacy (DP) mathematically ensures the indistinguishability between the computing outcomes on two datasets with or without an arbitrary individual.

Definition 1 ((ε,δ)-differential privacy [14]). *A randomized mechanism $\mathcal{M}$ satisfies (ε,δ)-DP, if for any two neighboring datasets $D \simeq D'$ and all $S \subseteq Range(\mathcal{M})$, it holds that*

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(\varepsilon)\Pr[\mathcal{M}(D') \in S] + \delta. \quad (8)$$

In Definition 1, the neighboring datasets D and D' can be transformed into each other by adding or removing a single item, ε is the privacy budget bounding the divergence of the output distribution between two neighboring datasets, and δ is negligible on the number of items in D measuring the probability that the above constraint is violated. When $\delta = 0$, (ε,δ)-DP becomes ε-DP, which is the first and strict definition of DP introduced by Dwork et al. [15].

Generally, a common approach to achieving ε-DP or (ε,δ)-DP is to inject noise following Laplace or Gaussian distribution into the original input/output/computing process. The magnitude of noise injection is related to the sensitivity of the function, f, for the computation. We provide the definition of ℓ_2-sensitivity as below

Definition 2 (ℓ_2-sensitivity [15]). *Given a function $f : \mathcal{D} \to \mathbb{R}^k$, for all $D \in \mathcal{D}$, the ℓ_2-sensitivity of f is*

$$S_f := \max_{D \simeq D'} \|f(D) - f(D')\|_2, \quad (9)$$

where $\|\cdot\|_2$ denotes ℓ_2-norm of the vector.

In practice, we say $\tilde{f}$ satisfies (ε,δ)-DP if we inject scaled Gaussian noise $\mathcal{N}(0,S_f^2\sigma^2\boldsymbol{I})$ into each component in a non-private f, where S_f is the ℓ_2-sensitivity of f and $\sigma \geq \sqrt{2\ln(1.25/\delta)}/\varepsilon$. The Gaussian mechanism is

$$\tilde{f}(D) := f(D) + \mathcal{N}(0,S_f^2\sigma^2\boldsymbol{I}). \quad (10)$$

Dong et al. [12] propose a variant of DP named Gaussian Differential Privacy (GDP). Let ϕ be a rejection rule used for testing the hypothesis to distinguish between two distributions P and Q. The trade-off function $F(P,Q) : [0,1] \to [0,1]$ is defined as $F(P,Q)(\alpha) := \inf_\phi\{1-\mathbb{E}_Q[\phi] : \mathbb{E}_P[\phi] \leq \alpha\}$, where $\mathbb{E}_P[\phi]$ is Type I error, and $1-\mathbb{E}_Q[\phi]$ is Type II error. Let $G_\mu := F(\mathcal{N}(0,1),\mathcal{N}(\mu,1))$, where $\mu \geq 0$. In practice, G_μ has a close-form expression as $G_\mu(\alpha) = \Phi(\Phi^{-1}(1-\alpha)-\mu)$, where Φ denotes the Cumulative Distribution Function (CDF) of standard normal distribution and Φ^{-1} is the inversion of Φ. Based on the trade-off function F, we have the definition of GDP as follows.

Definition 3 (Gaussian differential privacy (GDP) [12]). *A randomized mechanism $\mathcal{M}$ is said to satisfy μ-GDP if any two neighboring datasets D and D'*

$$F(\mathcal{M}(D),\mathcal{M}(D')) \geq G_\mu. \quad (11)$$

For the Gaussian mechanism, GDP provides a more straightforward case compared to (ε,δ)-DP. Given a specific noise scale σ, the following theorem demonstrates that the privacy budget of the Gaussian mechanism under GDP:

Theorem 1 (Gaussian mechanism on GDP [12]). *Given a noise scale σ, the Gaussian mechanism satisfies $1/\sigma$-GDP.*

An important DP (including GDP) property is post-processing, which ensures the DP for post-processing of an output of a DP mechanism.

Proposition 1 (Post-processing of DP [16]). *For any data-independent function g defined over the image of the randomized mechanism $\mathcal{M}$, if $\mathcal{M}$ is (ε,δ)-DP, then $g \circ \mathcal{M}$ satisfies (ε,δ)-DP, where $\circ$ denotes the composition operation.*

When applying DP on machine learning area to train privacy-preserved models, differentially private stochastic gradient descent (DP-SGD) [1] is the most common technique, which injects DP noise into the gradients. This way, the trained model will be differentially private. Then, based on the post-processing, all predictions made by such a model will be differentially private as well. However, one challenge for DP-SGD is to bound the global sensitivity due to no limit on the size of the gradients. To overcome this issue, DP-SGD clips each gradient to bound the global sensitivity of gradients, then injects Gaussian noise into each clipped gradient. Note that this process can be viewed as a Gaussian mechanism.

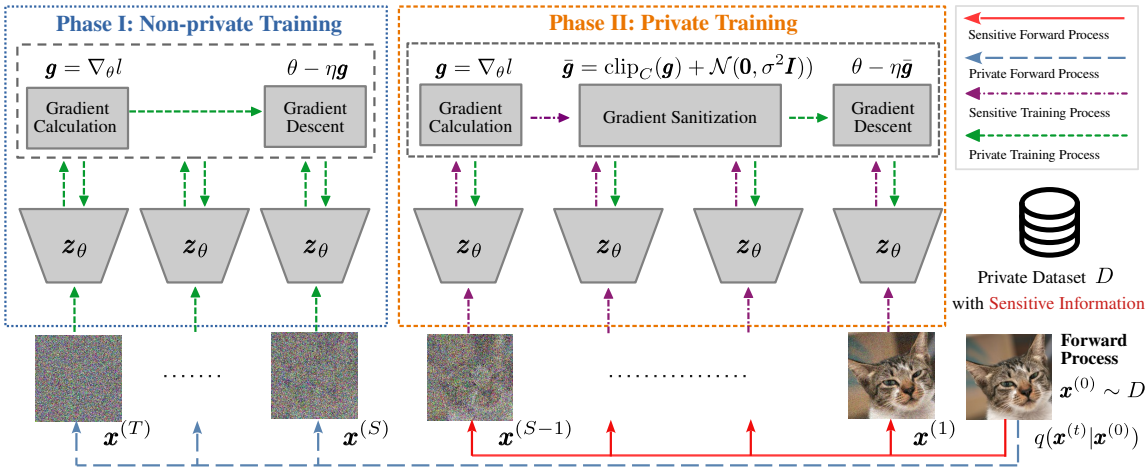


Figure 2: Framework of dp-promise, which aims to provide privacy guarantees to private data during the training of DMs. "Sensitive" means there is a risk of privacy leakage, and "Private" means privacy guarantees exist.

4 Threat Model

In this paper, we define white-box privacy adversaries against DMs as entities seeking to infer the existence of a particular image [13] or reconstruct a set of images [6] belonging to the DMs training data, given access to the images generated by DMs and the model parameters of the trained DMs. To preserve the privacy of the training data of DMs, we construct differentially private DMs, ensuring the synthetic data generated by DMs is differentially private.

Formally, we aim to train a differentially private diffusion model $\mathcal{G}$ with a neural network z_θ on a private dataset $D_{\text{priv}} = \{x_1, x_2, \ldots, x_n\}$. Given random noise r, the generated data by the DP DMs $\hat{x} \leftarrow \mathcal{G}(r)$ is differentially private. Note that we assume the model is publicly accessible, meaning that the adversaries have full access to both the diffusion model $\mathcal{G}$ and the corresponding neural network z_θ. We give the details of $\mathcal{G}$ and z_θ in Section 3.1.

Next, we provide the definition of the white-box membership inference.

Definition 4 (White-box membership inference attacks)**.** *Let $\mathcal{A}$ be a white-box adversary, $\mathcal{D}$ be data distribution, A be training algorithm, and $\mathcal{G}$ be a diffusion model with a neural network z_θ. The white-box membership inference attack is*

0. *$\mathcal{A}$ has full access to $\mathcal{G}$ and z_θ.*
1. *Select a private dataset $D_{priv} \in \mathcal{D}$.*
2. *Train $\mathcal{G}$ on D_{priv} with algorithm A as $\mathcal{G}_{A,D_{priv}} = A(\mathcal{G}, D_{priv})$.*
3. *Flip a coin to decide whether $b = 0$ or $b = 1$.*
4. *Sample $x \in D_{priv}$ if $b = 0$, $x \in \mathcal{D}$ if $b = 1$.*
5. *Attack is successful if $\mathcal{A}(x, \mathcal{G}_{A,D_{priv}}, \mathcal{D}) = b$, and fails otherwise.*

Note that DP limits the effect of any point in the private dataset to final computation results, thereby reducing the success rate of membership inference attacks [60].

5 Our Approach: dp-promise

In this section, we first introduce the motivation and propose a framework that ensures approximate DP in DMs. Then we describe the details of each component of dp-promise. Finally, we conduct a rigorous theoretical analysis for privacy guarantees of dp-promise.

5.1 Method Overview

The motivation of this work is mainly based on the observation that existing differentially private diffusion models [11, 17, 37] inject wasteful noise. That is, training DMs with DP-SGD injects DP noise on top of DM noise, which would result in duplicate noise injection and then damage the model utility. To address this problem, we propose Differentially Private Diffusion Probabilistic Models for Image Synthesis (dp-promise), which utilize DM noise when implementing DP. Figure 2 depicts the framework of dp-promise.

To take advantage of DM noise, we split the traditional DM training (recovering raw images by removing predicted noise) into two phases - Phase I non-private training (achieving DP by normal DM noise) and Phase II private training (achieving DP by DP-SGD). In Phase I, we follow the normal DM training during time-steps $[S, T]$. Recall Equation (4) in Section 3.1, the noisy images are produced by injecting scaled Gaussian noise, which could be treated as the Gaussian mechanism for achieving GDP. Then, based on the post-processing property, all following DM training operations in Phase I

are DP immediately. Proposition 2 gives a formal proof to connect the DM noise and DP guarantee. Following Phase I training, in Phase II, we apply DP-SGD to DM training during time-steps $[1, S-1]$ using DP-SGD to ensure DP guarantees directly. Overall, dp-promise ensures DP in the two phases while avoiding injecting DP noise during Phase I to save the privacy budget. Hence, dp-promise could provide better model utility without compromising the DP guarantee. Algorithm 1 describes the complete process of dp-promise.

Proposition 2. *Equation (4) satisfies* $2\sqrt{d\alpha_t/(1-\alpha_t)}$*-GDP.*

Proof. Let $\hat{\boldsymbol{x}} := \boldsymbol{x} + \mathcal{N}(\boldsymbol{0}, ((1-\alpha_t)/\alpha_t)\boldsymbol{I})$ be the procedure of generating noisy data in Equation (4). For any two data points $\boldsymbol{x}$ and $\boldsymbol{x}' \in \mathbb{R}^d$, we can derive that $\max_{\boldsymbol{x},\boldsymbol{x}'} \|\boldsymbol{x} - \boldsymbol{x}'\|_2 = 2\sqrt{d}$. Therefore, the sensitivity of the process is bounded by $2\sqrt{d}$. Following the Theorem 1, we can derive that the procedure of generating noisy data satisfies $2\sqrt{d\alpha_t/(1-\alpha_t)}$-GDP. □

Next, we give an example to show how dp-promise achieves DP on an image dataset. In this paper, we consider processing within the domain of images. Let $D_{\text{priv}} = \{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$ represent a private dataset, where we define the number of channels, height, and width of an image as *channel*, *height* and *width*, respectively. Consequently, $d = channel \times height \times width$ represents the dimensions of an image. Following the configuration detailed in [22], we employ a linear transformation for each pixel in the image, converting integer values within the range of $\{0, 1, \ldots, 255\}$ to the interval $[-1, 1]$.

Remark 1. *In practice, given a time-step* $t \in \{1, 2, \ldots, T\}$*,* $\{\alpha_1, \alpha_2, \ldots, \alpha_T\}$ *is a pre-defined decreasing sequence such that* α_1 *is close to* 1 *and* α_T *is close to* 0*. (e.g., with parameters like* $T = 1{,}000$*,* $\{\beta_1, \beta_2, \ldots, \beta_T\}$ *is selected from* 10^{-4} *to* 2×10^{-2} *linearly). Considering a high-dimensional image domain (e.g., a color image with* 32×32 *resolution and* 3 *channels, resulting in* $d = 3{,}072$*). Following Proposition 2, we can achieve significant DP guarantees when* $t \to T$ *and* $\gamma_t \to 0$ *(e.g., with parameters following above, resulting in* $\alpha_t = 0.0004$ *when* $t = 900$*). In contrast, when* $t \to 0$*,* $\gamma_t \to \infty$*, there is almost no privacy guarantee.*

In line with the analysis in Remark 1, it is evident that substantial DP guarantees are already established at the large time-steps.

5.2 Method Details

In this section, we give a detailed description of the two phases of dp-promise, then introduce several techniques applied in dp-promise for privacy and utility enhancement.

Two-phase training in dp-promise. We split the whole DMs training process (from time-steps 1 to T) at a given time-step $S \in \{1, 2, \ldots, T\}$. Since the key in the DMs training is to learn and remove the injected noise to generate the original image from a group of noisy images with different amounts of noise injected, we define Phase I as time-step $[S, T]$ and Phase II as time-step $[1, S-1]$.

Algorithm 1: dp-promise framework.

Input: private dataset $D_{\text{priv}} = \{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$, diffusion steps T, time-step boundary S, learning rate η_1 and η_2, batch size m_1 and m_2, number of iterations N_1 and N_2, clipping constant C, neural network $\boldsymbol{z}_\theta$, DP-SGD noise scale σ, number of samples for noise augmentation K.

// Phase I: training non-privately during time-step $[S, T]$

1 **for** *number of iterations* N_1 **do**
2 $\quad I \sim \texttt{PoissonSample}_{m_1/n}(\{1, 2, \ldots, n\})$
3 $\quad$ **for** $i \in I$ **do**
4 $\quad\quad t_i \sim \mathcal{U}(\{S, S+1, \ldots, T\}), \boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
5 $\quad\quad l_i \leftarrow \|\boldsymbol{z}_i - \boldsymbol{z}_\theta(\sqrt{\alpha_t}\boldsymbol{x}_i + \sqrt{1-\alpha_t}\boldsymbol{z}_i, t_i)\|_2^2$
6 $\quad \theta \leftarrow \theta - \eta_1 \cdot \frac{1}{|I|} \sum_{i \in I} \nabla_\theta l_i$

// Phase II: training privately during time-step $[1, S-1]$

7 **for** *number of iterations* N_2 **do**
8 $\quad I \sim \texttt{PoissonSample}_{m_2/n}(\{1, 2, \ldots, n\})$
9 $\quad$ **for** $i \in I$ **do**
10 $\quad\quad$ **for** $k = 1, 2, \ldots, K$ **do**
11 $\quad\quad\quad t_{ik} \sim \mathcal{U}(\{1, 2, \ldots, S-1\}), \boldsymbol{z}_{ik} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
12 $\quad\quad\quad l_{ik} \leftarrow \|\boldsymbol{z}_{ik} - \boldsymbol{z}_\theta(\sqrt{\alpha_t}\boldsymbol{x}_i + \sqrt{1-\alpha_t}\boldsymbol{z}_{ik}, t_{ik})\|_2^2$
$\quad\quad$ *// Aggregate and clip gradient*
13 $\quad\quad \bar{\boldsymbol{g}}_i \leftarrow \text{clip}_C(\frac{1}{K}\nabla_\theta \sum_{k=1}^K l_{ik})$
$\quad$ *// Add Gaussian noise to gradient*
14 $\quad \theta \leftarrow \theta - \eta_2 \cdot \frac{1}{|I|}(\sum_{i \in I} \bar{\boldsymbol{g}}_i + C\sigma \cdot \boldsymbol{\xi}), \boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

Output: trained parameters θ and output privacy budget by Theorem 2.

In Phase I, the main purpose is to leverage the injected DM noise to achieve DP. In the implementation, we first use Poisson sub-sampling to construct a mini-batch with probability m_1/n, where m_1 is the batch size of Phase I (step 2 of Algorithm 1). Then, we follow the normal DM training process to learn the noise injection distribution (steps 3 to 6 of Algorithm 1, please refer to Section 3 for details).

In Phase II, since the amount of DM-injected noise is not enough to achieve DP, we further apply DP-SGD [1] to guarantee DP. Formally, we use Poisson sub-sampling to construct a mini-batch with probability m_2/n, where m_2 is the batch size of Phase II. For $\boldsymbol{x}_i \sim D$, $t_i \sim \mathcal{U}(\{1, 2, \ldots, S-1\})$, and $\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, the parameters θ are updated as follows

$$\theta \leftarrow \theta - \eta_2 \cdot \frac{1}{|I|}\left(\sum_{i \in I} \text{clip}_C(\nabla_\theta l_i) + C\sigma \cdot \boldsymbol{\xi}\right), \qquad (12)$$

where $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, η_2 is learning rate in Phase II, and σ

is DP-SGD noise scale. The clipping function is defined as $\text{clip}_C(\boldsymbol{g}) = \boldsymbol{g}/\max\{1, \|\boldsymbol{g}\|_2/C\}$, where the clipping constant C (> 0) controls the magnitude of the gradient norm.

Dockhorn et al. [11] state that injecting extra noise into gradients leads to an increase in the variance of the gradient norm, resulting in a notable information loss after clipping gradients. Inspired by this point, we utilize the average per-instance loss computed over K times. That is, for each data point $\boldsymbol{x}_i$, we calculate the loss as $\tilde{l}_i = \frac{1}{K}\sum_{k=1}^{K} l_{ik}$, where $l_{ik} = \|\boldsymbol{z}_{ik} - \boldsymbol{z}_\theta(\sqrt{\alpha_t}\boldsymbol{x}_i + \sqrt{1-\alpha_t}\boldsymbol{z}_{ik}, t_{ik})\|_2^2$, $t_{ik} \sim \mathcal{U}(\{1,2,\ldots,S-1\})$, and $\boldsymbol{z}_{ik} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$. In practice, we calculate the gradient $\boldsymbol{g}_{ik} = \nabla_\theta l_{ik}$ at each iteration and then use the average to estimate per-instance gradients as $\nabla_\theta l_i = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{g}_{ik}$.

Privacy amplification by sub-sampling. During the training of differentially private DL models, sub-sampling is a common technique to enhance the DP guarantees [3, 31, 55, 65]. Specifically, sub-sampling involves applying the randomized mechanism to select a subset rather than the entire set. Since some instances are excluded from the chosen subset, sub-sampling prevents the privacy leakage of sensitive information associated with these excluded instances. dp-promise utilizes Poisson sub-sampling to select the mini-batch for Phase I and Phase II. Following the description in [55], we denote a sub-sampling procedure with a probability p as $\texttt{PoissonSample}_p$, and the definition is

Definition 5 (Poisson sub-sampling)**.** *Let S denote a set. $\texttt{PoissonSample}_p$ selects a subset from S. For each element in S, the selection is determined by an independent Bernoulli trial with probability $0 < p < 1$.*

By utilizing Poisson sub-sampling to construct the mini-batch at each training step, we can get privacy amplification through sub-sampling, reducing the overall privacy budget in two phases. We provide a detailed analysis of how sub-sampling amplifies privacy in Lemma 3, Section 5.3.

Pre-training on public data. Pre-training is a widely used strategy to improve the performance of DP generative models [17, 19, 57]. Intuitively, we assume that the public dataset used in pre-training does not contain any sensitive information. Thus, training on the public dataset does not incur privacy costs. Let D_{pub} represent a publicly available dataset. We denote the pre-training integrated process as $\mathcal{M} = \mathcal{M}_{D_{\text{priv}}} \circ \mathcal{M}_{D_{\text{pub}}}$, where we first pre-train on D_{pub} and then fine-tune on D_{priv}. The privacy guarantee of $\mathcal{M}$ is determined by the privacy guarantee of $\mathcal{M}_{D_{\text{priv}}}$. In dp-promise, the consumed privacy budgets are directly related to the number of training iterations since each iteration will consume an amount of privacy budget (i.e., more training iterations incur a higher privacy budget). Hence, we employ pre-training to accelerate the training process and reduce the number of training iterations. Additionally, even in the presence of a noticeable distribution shift between the public and private datasets, the experiments have demonstrated that pre-training remains effective. In practice, the time-step, as an input to the network, is encoded in an embedding layer with learnable parameters. Training the entire network with DP-SGD would introduce unnecessary noise into these parameters. Therefore, we freeze these layers before the fine-tuning process to reduce information loss.

Algorithm 2: Sampler.

Input: sampling step T', variance hyper-parameter ρ, trained neural network $\boldsymbol{z}_\theta$, target label y (for conditional generation).

$\hat{x}^{(\tau_{T'})} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$

for $\tau_t = \tau_{T'}, \tau_{T'-1} \ldots, \tau_1$ **do**

  $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$ if $t > 1$ else $\boldsymbol{\xi} = 0$

  if *unconditional generation* **then**

$$\hat{\boldsymbol{x}}^{(\tau_{t-1})} = \sqrt{\alpha_{\tau_{t-1}}}\frac{\hat{\boldsymbol{x}}^{(\tau_t)} - \sqrt{1-\alpha_{\tau_t}}\boldsymbol{z}_\theta(\hat{\boldsymbol{x}}^{(\tau_t)}, \tau_t)}{\sqrt{\alpha_{\tau_t}}} + \sqrt{1-\alpha_{\tau_{t-1}} - \hat{\Sigma}_{\tau_t}^2}\cdot\boldsymbol{z}_\theta(\hat{\boldsymbol{x}}^{(\tau_t)}, \tau_t) + \hat{\Sigma}_{\tau_t}\boldsymbol{\xi}$$

  else if *conditional generation* **then**

$$\hat{\boldsymbol{x}}^{(\tau_{t-1})} = \sqrt{\alpha_{\tau_{t-1}}}\frac{\hat{\boldsymbol{x}}^{(\tau_t)} - \sqrt{1-\alpha_{\tau_t}}\hat{\boldsymbol{z}}_\theta(\hat{\boldsymbol{x}}^{(\tau_t)}, \tau_t, y)}{\sqrt{\alpha_{\tau_t}}} + \sqrt{1-\alpha_{\tau_{t-1}} - \hat{\Sigma}_{\tau_t}^2}\cdot\hat{\boldsymbol{z}}_\theta(\hat{\boldsymbol{x}}^{(\tau_t)}, \tau_t, y) + \hat{\Sigma}_{\tau_t}\boldsymbol{\xi}$$

Output: synthetic sample $\hat{\boldsymbol{x}}^{(\tau_0)}$.

Sampling. dp-promise leverages DDIM sampler [48] to accelerate the sampling process in the DM training. Given a sample step T', we linearly choose a subset $\{\tau_1, \tau_2, \ldots, \tau_{T'}\}$ from the complete time-steps set. In unconditional generation, we start with a random noisy data point $\hat{\boldsymbol{x}}^{\tau_{T'}} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$ and iteratively generate synthetic data from $\tau_{T'-1}$ to τ_0 by Equation (7). In conditional generation, we employ classifier-free guidance [23] in the sampling process of DMs.

For downstream tasks involving classifier training, it is necessary to utilize label information as a condition to guide the training process. Nevertheless, if original labels are directly inputted in Phase I, the label information will propagate through gradients into the model due to no additional noise added to gradients, which may pose a risk of privacy leakage. To address this issue, we do not use input label information in Phase I but use the labels in Phase II. Since the gradient information is perturbed by additional noise in Phase II, we cannot directly propagate the private information in the labels into the model. During the sampling process, the role of labels is to guide the model to generate data for the corresponding label. We utilize both the conditional and unconditional prediction results with a guidance scale w as $\hat{\boldsymbol{z}}_\theta(\boldsymbol{x}^{(t)}, t, y) = (1+w)\boldsymbol{z}_\theta(\boldsymbol{x}^{(t)}, t, y) - w\boldsymbol{z}_\theta(\boldsymbol{x}^{(t)}, t)$. The entire sampling procedure is presented in Algorithm 2.

5.3 Privacy Analysis

To calculate the overall privacy budget consumption, we provide a theoretical analysis for dp-promise. In this paper, we employ the Gaussian differential privacy (GDP) [12] as the foundation for theoretical analysis and utilize sub-sampling technology to amplify the DP guarantees. As a result, we provide an approximate bound of the privacy budget under (ε, δ)-DP for Algorithm 1.

Next, we introduce the conversion between GDP and (ε, δ)-DP. Specifically, for any fixed probability δ, μ-GDP can be transformed into $(\varepsilon_{\delta,\mu}, \delta)$-DP, where $\varepsilon_{\delta,\mu}$ is constrained by a condition detailed in the following lemma

Lemma 1 (GDP to DP conversion [12]). *A randomized mechanism $\mathcal{M}$ is μ-GDP if and only if $\mathcal{M}$ is $(\varepsilon, \delta(\varepsilon))$-DP for all $\varepsilon \geq 0$, where*

$$\delta(\varepsilon) = \Phi(-\frac{\varepsilon}{\mu} + \frac{\mu}{2}) - \exp(\varepsilon)\Phi(-\frac{\varepsilon}{\mu} - \frac{\mu}{2}). \tag{13}$$

Given a randomized mechanism $\mathcal{M}$, which is assembled as the composition of k randomized mechanisms $\mathcal{M}_i, \mathcal{M}_{i+1}, \ldots, \mathcal{M}_k$. If each $\mathcal{M}_i$ satisfies GDP, it naturally follows that $\mathcal{M}$ also maintains GDP. Formally, we provide the composition of GDP as follows

Lemma 2 (Composition on GDP [12]). *The k-fold composition of μ_i-GDP $(0 \leq i \leq k)$ mechanisms satisfy $\sqrt{\mu_1^2 + \cdots + \mu_k^2}$-GDP.*

dp-promise utilizes a sub-sampling technique to strengthen DP guarantees. Specifically, we employ Poisson sub-sampling to select a subset from the dataset, and then we apply a private algorithm to this selected subset. Next, we describe the privacy amplification provided by GDP as follows

Lemma 3 (Sub-sampling on GDP [3]). *Let a randomized mechanism $\mathcal{M}$ satisfy μ-GDP, and $\mathcal{M} \circ \texttt{PoissonSample}_p$ denote the privacy amplification by Poisson sub-sampling with probability p. Suppose p is related to N such that $p\sqrt{N} \to v$, the N-fold composition of $\mathcal{M} \circ \texttt{PoissonSample}_p$ asymptotically satisfies $\hat{\mu}$-GDP as $N \to \infty$, where*

$$\hat{\mu} = v\sqrt{\exp(\mu^2) - 1} = p\sqrt{N(\exp(\mu^2) - 1)}. \tag{14}$$

For any randomized mechanism $\mathcal{M}$ that satisfies μ-GDP, we can apply $\texttt{PoissonSample}_p$ to obtain a new mechanism denoted as $\mathcal{M} \circ \texttt{PoissonSample}_p$. Following Lemma 3, we can derive the new mechanism that satisfies $\hat{\mu}$-GDP. To ensure DP guarantees for both Phase I and Phase II, we present Lemma 4 and Lemma 5, respectively.

Lemma 4. *Given a time-step boundary S for splitting Phase I and Phase II, a batch size m_1, the size of the private dataset n, the data dimensions d, the pre-defined diffusion noise scale α_S, and the number of iterations N_1, Phase I in Algorithm 1 asymptotically satisfies μ_1-GDP, where*

$$\mu_1 = \frac{m_1}{n}\sqrt{N_1(\exp(4d\alpha_S/(1-\alpha_S)) - 1)}. \tag{15}$$

Proof. Let $\mathcal{M}_1$ denote the mechanism in Phase I such that $\mathcal{M}_1(\boldsymbol{x}) := \boldsymbol{x} + \mathcal{N}(\mathbf{0}, ((1-\alpha_t)/\alpha_t)\boldsymbol{I})$, where $t \in \{S, S+1, \ldots, T\}$. Following Proposition 2, we can derive that $\mathcal{M}_1$ satisfies $2\sqrt{d\alpha_S/(1-\alpha_S)}$-GDP. For a private dataset D, we denote the sequence of $\mathcal{M}_1(\boldsymbol{x}_i)$ as $\mathcal{M}_1(D) := D + \mathcal{N}((1-\alpha_t)/\alpha_t)\boldsymbol{I})$. For any two neighboring datasets $D = \{\boldsymbol{x}_i\}_{i=1}^n \cup \{\boldsymbol{x}'\}$ and $D' = \{\boldsymbol{x}_i\}_{i=1}^n$, where $\boldsymbol{x}' \notin D'$, the sensitivity can be bounded as $\max_{D,D'} \|D - D'\| = 2\sqrt{d}$. Following Theorem 1, the sequence of $\mathcal{M}_1(\boldsymbol{x}_i)$ satisfies $2\sqrt{d\alpha_S/(1-\alpha_S)}$-GDP. Moreover, we use Poisson sub-sampling to amplify differential privacy. Following Lemma 3, given sub-sampling probability $p_1 = m_1/n$, Phase I asymptotically satisfies $p_1\sqrt{N_1(\exp(4d\alpha_S/(1-\alpha_S)) - 1)}$-GDP. □

In Phase I, α_S is a constant related to the selection of S. As S increases, the value of α_S decreases. In particular, $\alpha_t = \prod_{s=1}^t (1-\beta_s)$, where we set β_t to linearly increase from $\beta_1 = 10^{-4}$ to $\beta_T = 2 \times 10^{-2}$. For different data dimensions, as the data dimension d increases, a larger value of S is needed to ensure practical privacy guarantees. In contrast, for low-dimensional data, practical privacy guarantees can be achieved without selecting larger values of S. We show the selection for the value of S in Section 6.

Lemma 5. *Given a DP-SGD noise scale σ, a batch size m_2, the size of the private dataset n, and the number of iterations N_2, Phase II in Algorithm 1 satisfies μ_2-GDP, where*

$$\mu_2 = \frac{m_2}{n}\sqrt{N_2(\exp(1/\sigma^2) - 1)}. \tag{16}$$

Proof. Let $\mathcal{M}_2$ denote the mechanism in Phase II (based on Equation (12)) such that $\mathcal{M}_2(D) = g(D) + C\sigma \cdot \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, where $g(D) = \sum_{i \in I} \text{clip}_C(\nabla_\theta l_i)$. For two neighboring datasets $D = \{\boldsymbol{x}_i\}_{i=1}^n \cup \{\boldsymbol{x}'\}$ and $D' = \{\boldsymbol{x}_i\}_{i=1}^n$, where $\boldsymbol{x}' \notin D'$, the sensitivity of g is

$$\begin{aligned}
&\max_{D,D'} \left\| g(D) - g(D') \right\|_2 \\
=&\max_{D,D'} \left\| \sum_{i=1}^n \text{clip}_C(\nabla_\theta l_i) + \text{clip}_C(\nabla_\theta l') - \sum_{i=1}^n \text{clip}_C(\nabla_\theta l_i) \right\|_2 \\
=&\max_{D,D'} \left\| \text{clip}_C(\nabla_\theta l') \right\|_2 = C,
\end{aligned}$$

where $\nabla_\theta l'$ is the gradient with respect to $\boldsymbol{x}'$. Following Theorem 1, $\mathcal{M}_2$ satisfies $1/\sigma$-GDP. Subsequently, we use Poisson sub-sampling to amplify differential privacy. Following Lemma 3, given sub-sampling probability $p_2 = m_2/n$, Phase II asymptotically satisfies $p_2\sqrt{N_2(\exp(1/\sigma^2) - 1)}$-GDP. □

In Phase II, the noise scale σ controls the magnitude of the injected noise (with more noise injected providing greater privacy guarantees for the DMs). The clipping constant C controls the sensitivity of gradients. Changing the clipping constant will not affect the privacy guarantees when other parameters remain unchanged. This is because adjusting the clipping constant simultaneously affects both the overall noise added in Phase II and the sensitivity of gradients, thereby maintaining the final privacy guarantee unchanged.

Following the composition property on GDP (Lemma 2), we compose the overall privacy budget consumption of the two phases, as detailed in Lemma 4 and Lemma 5, respectively. Additionally, we employ the conversion to convert the privacy budget from GDP to (ε, δ)-DP for comparative analysis. Next, we provide the formal privacy guarantees for dp-promise as follows

Theorem 2 (Differential privacy for dp-promise)**.** *Algorithm 1 asymptotically satisfies $(\varepsilon, \delta(\varepsilon))$-DP, it holds that*

$$\delta(\varepsilon) = \Phi(-\frac{\varepsilon}{\mu} + \frac{\mu}{2}) - \exp(\varepsilon)\Phi(-\frac{\varepsilon}{\mu} - \frac{\mu}{2}), \tag{17}$$

$$\mu = \sqrt{\mu_1^2 + \mu_2^2}, \tag{18}$$

where μ_1 is defined in Equation (15) and μ_2 is defined in Equation (16).

Proof. Algorithm 1 contains two phases. Following composition on GDP via Lemma 2, we can derive that Algorithm 1 asymptotically satisfies $\sqrt{\mu_1^2 + \mu_2^2}$-GDP. Then, following Lemma 1, we can obtain the final privacy budget $(\varepsilon, \delta(\varepsilon))$. □

According to the condition provided in Theorem 2, in practice, we can also calculate a privacy budget ε_δ by a given δ. Note that the results we obtained from the theoretical analysis are approximate compositions based on the Central Limit Theorems (CLT) of f-DP [12]. Nevertheless, Dong et al [12] and Bu et al. [3] demonstrate that the approximation derived from CLT is close to the exact composition results in their experiments. Furthermore, two commonly used DP-ML frameworks, i.e., TensorFlow Privacy[1] and Opacus [61], both support the implementation of the GDP accountant, which is based on the approximate results from CLT [3, 12]. Additionally, Dong et al. [12] emphasized that the exact computation of privacy guarantees under compositions is computationally hard, thus tractable approximations are important. Eventually, we utilize the differentially private DMs to generate synthetic data. The following corollary of Theorem 2 presents the DP guarantees for synthetic data generated by dp-promise.

Corollary 1. *Given a private dataset D_{priv} and a fixed probability δ, let $\mathcal{G}$ represent the diffusion model, such that the neural network z_θ trained by Algorithm 1. Let $X = \mathcal{G}(R)$ represent the diffusion model map a noise set $R \in \mathcal{R}$ to a synthetic dataset $X \in \mathcal{X}$, where $\mathcal{R}$ and $\mathcal{X}$ are noise space and data space, respectively. For any $R \in \mathcal{R}$, there is $\varepsilon_\delta > 0$ such that the synthetic dataset asymptotically satisfies $(\varepsilon_\delta, \delta)$-DP.*

Proof. The noise set R is independent of the private dataset D_{priv}. Following the Theorem 2, the privacy in parameters of the neural network is asymptotically bounded by $(\varepsilon_\delta, \delta)$-DP. Additionally, there is no data-dependent parameter in the sampling procedure of DMs. According to the post-processing property of DP, the synthetic dataset generated by $\mathcal{G}$ asymptotically satisfies $(\varepsilon_\delta, \delta)$-DP. □

6 Experimental Evaluation

In this section, we evaluate dp-promise to demonstrate the performance in generating differentially private data. To show the effectiveness of dp-promise, we compare dp-promise against the state-of-the-art differentially private generative models on image data.

6.1 Experiment Setup

Datasets. Our experiments are conducted on four well-known image datasets for comprehensive evaluation, i.e., MNIST [30], Fashion-MNIST [58], CelebA [35], and CIFAR-10 [29]. Specifically, MNIST [30] and Fashion-MNIST [58] are two widely used datasets in the field of differentially private image synthesis. Both MNIST and Fashion-MNIST contain a total of 70,000 grayscale images, each with a resolution of 28×28 pixels, comprising handwritten digits and fashion products with 10 distinct object classes, respectively. MNIST and Fashion-MNIST are divided into two parts: 60,000 images for training and 10,000 images for testing. In the experiments, 60,000 training images are utilized in the training procedure. Furthermore, we also investigate two more complex and high-dimensional datasets, namely CelebA [35] and CIFAR-10 [29]. CelebA contains 202,599 color images with a resolution of 64×64 pixels, and CIFAR-10 comprises 60,000 color images, each with a resolution of 32×32 pixels.

Baselines. Our proposed method, dp-promise, is compared with various baseline models, including DP-SGD DM, DP-MERF [18], DPDM [11], DP-MEPF [19], and DP-Diffusion [17]. Note that DPGEN [8] is excluded from the comparison due to an incorrect privacy analysis [11]. PATE-based methods (e.g., G-PATE [36]) rely on data-dependent privacy, thereby making the PATE-based methods incomparable to dp-promise.

- **DP-SGD DM**: This is a naive method where DP-SGD is directly applied to fine-tuning a pre-trained DM. For DP-SGD DM, we first pre-train DMs using public data and then fine-tune the DMs with private data. Note that

[1] https://github.com/tensorflow/privacy

Figure 3: The synthetic data generated by DP-MERF, DPDM, DP-MEPF, DP-SGD DM, and dp-promise under $\varepsilon = 10$ and $\delta = 10^{-5}$ on MNIST and Fashion-MNIST. The original data is presented in the last row.

we maintain the model architecture and parameters as close to dp-promise as possible.

- **DP-MERF [18] & DP-MEPF [19]**: We executed the official code provided by the authors of DP-MERF[2] and DP-MEPF[3] to obtain the results. For DP-MEPF, we consider both features ϕ_1 and ϕ_2 within DP-MEPF. Note that DP-MEPF utilizes public data.
- **DPDM [11]**: We executed the official code[4] provided by the authors. Note that the original DPDM conducted experiments without public data pre-training and had two distinct settings - DPDM (FID) and DPDM (Acc), which focus on sample quality and downstream utility, respectively. For fair comparisons of experiments with public data pre-training, we made necessary modifications, such as parameters based on the official code to adapt pre-training settings, denoted as DPDM (Pub).
- **DP-Diffusion [17]**: Due to no publicly available code of DP-Diffusion [17] and no experimental results on MNIST, Fashion-MNIST, and CelebA datasets, we manually reproduced DP-Diffusion following all provided information from [17] and reported the results on the high-dimensional CelebA dataset and the more challenging CIFAR-10 dataset.

Evaluation metrics. To demonstrate the performance of dp-promise in differentially private synthetic image generation, we conduct a comprehensive analysis using both sample quality and downstream utility. To compare sample quality with other approaches, we present quantitative results that include the Fréchet Inception Distance (FID) [20] and Inception Score (IS) [45]. These two metrics are commonly used to evaluate the quality and diversity of images generated by the generative model. Informally, FID is computed based on the distance between the feature distributions extracted from generated images and real images, and the Inception Score is calculated as the exponential mean of the probability distribution extracted from generated images. To measure the utility of synthetic data in downstream tasks, following existing studies [5, 11, 18, 24], we utilize a range of classifiers, including multi-layer perceptron (MLP), convolutional neural network (CNN), and eleven additional scikit-learn [28] classifiers (e.g., logistic regression, decision tree, etc.). In particular, we train each classifier on synthetic data and then evaluate the trained classifiers on real data to measure the performance in downstream tasks using classification accuracy. We report the accuracy of MLP, CNN, and the average accuracy of the other eleven scikit-learn classifiers.

6.2 Implementation

Model architectures and implementation. In the experiments, we use the PyTorch framework [43]. Specifically, we employ the Opacus framework [61] to implement the DP-SGD algorithm. Then, we build neural networks us-

[2] https://github.com/ParkLabML/DP-MERF
[3] https://github.com/ParkLabML/DP-MEPF
[4] https://github.com/nv-tlabs/DPDM

Table 2: This table displays the downstream utility and sample quality of synthesized data generated by DP-MERF, DP-MEPF, DPDM, DP-SGD DM, and dp-promise under different privacy budgets ε and $\delta = 10^{-5}$. The metrics include MLP classifier accuracy (MLP%), CNN classifier accuracy (CNN%), the average of 11 scikit-learn classifiers accuracy (Avg%), and FID.

MNIST	D_{pub}	$\varepsilon = \infty$ (Non-private)				$\varepsilon = 10$				$\varepsilon = 1$				$\varepsilon = 0.2$			
		MLP	CNN	Avg	FID↓	MLP	CNN	Avg	FID↓	MLP	CNN	Avg	FID↓	MLP	CNN	Avg	FID↓
DP-MERF [18]	✗	80.4	83.5	70.5	104.4	80.0	83.5	68.6	105.6	80.0	82.3	66.3	110.9	76.2	79.0	58.2	133.3
DPDM (FID) [11]	✗	95.7	98.6	85.7	2.0	94.5	97.8	85.4	4.4	87.7	92.7	77.8	22.4	66.4	71.2	54.1	60.8
DPDM (Acc) [11]	✗	**96.6**	**98.9**	**86.4**	1.9	95.2	98.0	**85.8**	5.9	91.5	95.1	82.1	34.1	78.0	84.6	71.6	101.9
DP-MEPF [19]	✓	87.6	94.3	77.9	167.2	87.8	94.3	77.5	167.0	87.2	93.7	75.3	166.3	76.5	85.7	58.3	180.2
DP-SGD DM	✓	96.4	98.6	86.2	1.7	94.5	97.6	85.1	3.0	90.8	94.1	75.5	8.6	56.8	65.3	42.8	28.3
DPDM (Pub)	✓	96.5	98.8	**86.4**	1.9	95.3	97.8	85.6	3.9	92.3	95.6	82.2	9.0	81.3	86.2	**73.3**	26.5
dp-promise (this work)	✓	96.4	98.7	86.1	**1.6**	**95.9**	**98.2**	85.6	**2.3**	**93.6**	**95.8**	**83.0**	**6.6**	**84.8**	**87.6**	72.3	**23.1**

Fashion-MNIST	D_{pub}	$\varepsilon = \infty$ (Non-private)				$\varepsilon = 10$				$\varepsilon = 1$				$\varepsilon = 0.2$			
		MLP	CNN	Avg	FID↓	MLP	CNN	Avg	FID↓	MLP	CNN	Avg	FID↓	MLP	CNN	Avg	FID↓
DP-MERF [18]	✗	73.8	63.4	63.2	103.3	72.6	70.0	60.6	100.7	75.1	64.0	58.7	96.5	70.6	69.0	52.4	149.8
DPDM (FID) [11]	✗	84.8	87.3	**74.1**	8.0	82.6	85.3	72.1	17.9	74.4	77.1	66.7	45.1	55.3	55.5	45.6	76.7
DPDM (Acc) [11]	✗	86.4	87.7	73.3	7.0	83.1	85.4	72.6	18.1	76.1	78.6	68.8	50.3	69.2	72.7	65.5	126.5
DP-MEPF [19]	✓	74.9	79.4	69.7	86.7	74.0	78.7	66.0	89.1	74.5	76.7	63.2	102.3	**71.0**	69.7	47.1	167.5
DP-SGD DM	✓	85.8	87.6	73.8	5.7	82.3	84.6	71.1	6.4	65.7	69.7	53.9	16.5	44.2	50.8	41.7	38.4
DPDM (Pub)	✓	**86.5**	**87.9**	73.9	5.2	82.0	85.0	71.2	10.4	76.5	80.2	**69.8**	20.9	70.4	**73.8**	**68.3**	40.2
dp-promise (this work)	✓	85.7	87.4	73.5	**4.8**	**83.4**	**85.5**	**73.1**	**6.3**	**78.4**	**81.6**	69.2	**13.6**	67.8	68.5	62.4	**34.8**

ing a U-Net architecture, which is based on the improved DDPM [40] repository[5], and we train the network using Adam [27] optimizer. In dp-promise, we set diffusion steps to $T = 1,000$ and linearly assign values from 10^{-4} to 2×10^{-2} for $\{\beta_1, \beta_2, \ldots, \beta_T\}$. Similar to recent works [2, 11, 17], we use a large batch size and small clipping constant during the private training procedure to enhance the sample quality. We show the values of all the hyper-parameters in the "Training" paragraph. Following existing work [11, 49], we use the exponential moving average (EMA) of neural network parameters, which is a common practice in traditional DMs.

Training. For MNIST, we initiate pre-training of DMs with 1.6M parameters using Fashion-MNIST as the public dataset. The network contains 32 base channels, channel multipliers $(1,2,2)$, and attention resolution 7. We pre-train this network with a learning rate of 2×10^{-4} and 50 epochs at batch size 128. In the experiments on Fashion-MNIST, we pre-train DMs with 6.5M parameters on CIFAR-10. In this case, we transformed each image into grayscale and resized the images into 28×28 pixels, matching the size of Fashion-MNIST. The network has 64 base channels, channel multipliers $(1,2,2)$, and attention resolution 7 in the neural network. Pre-training is performed with a learning rate of 2×10^{-4} and $1,000$ epochs at batch size 128. Subsequently, we fine-tune these pre-trained models on MNIST and Fashion-MNIST using dp-promise. The fine-tuning process involves time-step boundary $S = 900$, learning rate $\eta_1 = 3 \times 10^{-4}, \eta_2 = 6 \times 10^{-4}$, batch size $m_1 = 32, m_2 = 4,096$, and the number of iterations $N_1 = 3 \cdot n/m_1, N_2 = 50 \cdot n/m_2$. We apply noise augmentation with $K = 32$ and a clipping constant of $C = 10^{-2}$. For the selection of S, due to the high-dimensional data we need to handle, it is necessary to choose a large S to ensure that Phase I does not consume an excessive amount of privacy budget. In the sampling procedure, we consider conditional generation with sampling parameters $T' = 200$, $\rho = 1$, and $w = 0$.

For CelebA and CIFAR-10 on 32×32 resolution, we initiate the pre-training of DMs with 35M parameters on ImageNet [9] as the public dataset, where each image is resized into 32×32 resolution, matching the dimensions of CelebA and CIFAR-10. The network includes 128 base channels, channel multipliers $(1,2,2,2)$, and attention resolution 16. Pre-training is performed with a learning rate of 2×10^{-4} and 50 epochs at batch size 128. For CelebA, we fine-tune the network using dp-promise with a time-step boundary $S = 925$, learning rate $\eta_1 = 3 \times 10^{-4}, \eta_2 = 3 \times 10^{-4}$, batch size $m_1 = 32, m_2 = 4,096$, the number of iterations $N_1 = 2 \cdot n/m_1, N_2 = 50 \cdot n/m_2$, noise augmentation $K = 4$, and clipping constant $C = 10^{-2}$. For CIFAR-10, we apply parameters including learning rate $\eta_1 = 3 \times 10^{-4}, \eta_2 = 3 \times 10^{-4}$, batch size $m_1 = 32, m_2 = 4,096$, the number of iterations $N_1 = 3 \cdot n/m_1, N_2 = 50 \cdot n/m_2$, noise augmentation $K = 4$, and clipping constant $C = 10^{-2}$. In the sampling procedure, we consider unconditional generation with sampling parameters $T' = 200$, $\rho = 1$, and $w = 0$.

Evaluation. For evaluation of sample quality, we use the code and Inception network at `tensorflow_gan` repository[6] to compute FID and Inception Score. We use 60,000 synthetic images and then compute the Inception Score and FID with the origin training dataset. For the evaluation of downstream quality, we build the MLP and CNN classifiers based on the

[5] https://github.com/openai/improved-diffusion

[6] https://github.com/tensorflow/gan

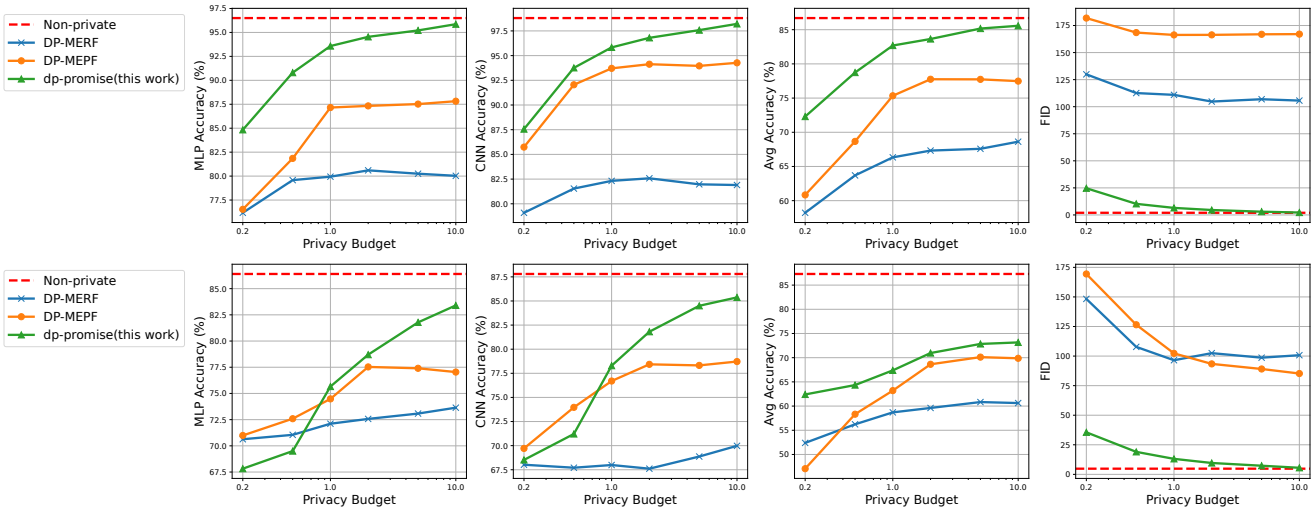
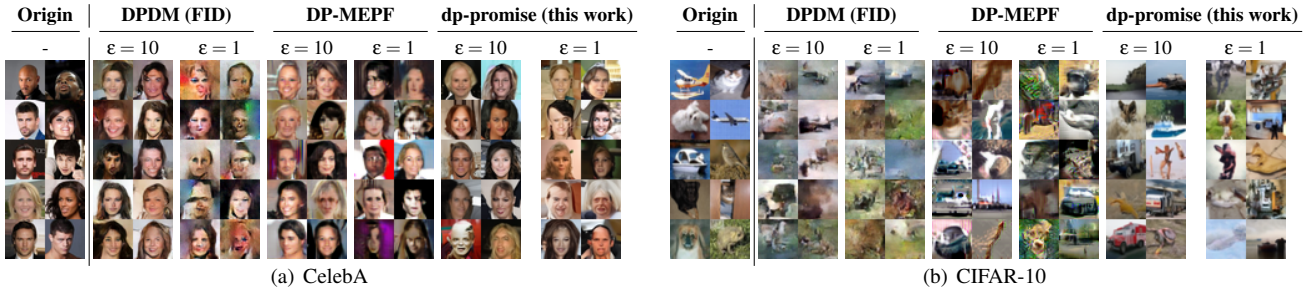


Figure 4: Privacy-utility trade-off comparison of DP-MERF, DP-MEPF, and dp-promise under various privacy budgets ε with fixed $\delta = 10^{-5}$ on MNIST (the first row) and Fashion-MNIST (the second row).



(a) CelebA (b) CIFAR-10

Figure 5: Synthetic data generated by DPDM, DP-MEPF, and dp-promise under various privacy budgets with fixed $\delta = 10^{-5}$ on CelebA (left) and CIFAR-10 (right). The original images are displayed in the first column for each dataset.

DPDM repository. On MNIST and Fashion-MNIST, we randomly split 60,000 synthetic samples into a 50,000 training set and a 10,000 validation set. Then we train all classifiers using Adam optimizer with 50 epochs, batch size 128, and a learning rate of 3×10^{-4}. Finally, we select the classifier that achieves the highest accuracy on the validation set and then test the classifier on the real data.

6.3 Experimental Results

MNIST and Fashion-MNIST. Following the previous literature on differentially private generative models [11, 18, 24], we compare the sample quality and downstream utility for dp-promise in comparison to existing methods. We perform on MNIST and Fashion-MNIST, considering fixed privacy budgets $\varepsilon = \{0.2, 1, 10, \infty\}$ with a fixed probability $\delta = 10^{-5}$, where $\varepsilon = \infty$ represents the non-private settings. Note that we vary privacy budgets ε from 0.2 to 10, as this range is commonly employed in practical applications like DP-FL [62, 63], DP-SGD [4, 46, 56], DP-DL [42], DPML [39] and Vote-Histogram [54].

In Figure 3, we present the visualization results of synthetic data generated by various approaches on both MNIST and Fashion-MNIST. We can observe that dp-promise generates high-fidelity samples under a standard privacy guarantee (e.g., $\varepsilon = 1$). The aggregated results for sample quality and downstream utility of synthetic data are summarized in Table 2. Specifically, on MNIST, dp-promise outperforms all baselines on almost all metrics in downstream utility and sample quality. On Fashion-MNIST, dp-promise achieves a lower FID compared to other approaches under all privacy budgets while achieving downstream prediction accuracy similar to DP-MEPF and DPDM under a strong privacy guarantee (e.g., $\varepsilon = 0.2$). For DPDM with public data settings, pre-training enables DPDM to enhance sample quality, but we find there is no significant improvement in downstream utility. dp-promise still outperforms DPDM with public data pre-training in most settings. Note that CIFAR-10 and Fashion-MNIST have a significant visual difference, making it challenging to trans-

Table 3: The sample quality of synthetic data generated by DPDM, DP-MEPF, DP-Diffusion, and dp-promise on CelebA and CIFAR-10 under various privacy budgets with $\delta = 10^{-5}$ and $\delta = 10^{-6}$, respectively.

CelebA	D_{pub}	$\varepsilon = 10$		$\varepsilon = 5$		$\varepsilon = 1$	
		FID↓	IS↑	FID↓	IS↑	FID↓	IS↑
DPDM (FID) [11]	✗	20.9	2.0	45.8	2.1	72.5	2.1
DP-MEPF [19]	✓	18.0	**2.5**	18.9	2.4	19.7	**2.6**
DPDM (Pub)	✓	8.6	**2.5**	8.8	2.4	10.4	2.4
DP-Diffusion [17]	✓	8.5	2.4	9.5	**2.6**	12.2	2.6
dp-promise (this work)	✓	**6.0**	**2.5**	**6.5**	2.5	**9.0**	**2.6**

CIFAR-10	D_{pub}	$\varepsilon = 10$		$\varepsilon = 5$		$\varepsilon = 1$	
		FID↓	IS↑	FID↓	IS↑	FID↓	IS↑
DPDM (FID) [11]	✗	92.8	3.7	106.5	3.5	128.4	3.4
DP-MEPF [19]	✓	32.6	7.3	38.8	6.5	43.2	6.1
DPDM (Pub)	✓	20.9	8.4	22.7	8.3	27.6	8.2
DP-Diffusion [17]	✓	19.8	8.2	23.5	8.1	26.5	8.5
dp-promise (this work)	✓	**17.9**	**8.6**	**18.9**	**8.7**	**21.8**	**9.1**

Table 4: The comparison of dp-promise with/without Phase I on MNIST and Fashion-MNIST under $\varepsilon = 10$ and $\delta = 10^{-5}$.

Methods	**MNIST**				**Fashion-MNIST**			
	MLP	CNN	Avg	FID↓	MLP	CNN	Avg	FID↓
without Phase I	95.7	97.8	84.4	2.5	82.2	83.5	**72.7**	6.8
with Phase I	**95.8**	**98.1**	**84.8**	**2.3**	**82.4**	**84.9**	72.5	**6.5**

fer knowledge under a large distribution shift. Nonetheless, dp-promise achieves higher downstream classification accuracy under standard privacy guarantees (e.g., $\varepsilon = \{1, 10\}$). In comparison with DPDM, dp-promise simultaneously achieves high sample quality and downstream utility across one setting, in contrast to the distinct settings in DPDM.

To explore the privacy-utility trade-off of different methods, we vary the privacy budget ε from 0.2 to 10 for DP-MERF, DP-MEPF, and dp-promise, and then report the downstream utility and sample quality on MNIST and Fashion-MNIST. As shown in Figure 4, under the standard privacy budgets (e.g., $\varepsilon = 1, 10$), we observe that dp-promise consistently outperforms all of the other approaches, and dp-promise shows lower sample quality degradation. Moreover, under a higher privacy budget, dp-promise achieves performance that is close to the non-private setting, highlighting the effectiveness in preserving both utility and privacy.

CelebA and CIFAR-10. To demonstrate the usability of dp-promise on more complex datasets beyond MNIST and Fashion-MNIST, we compare the sample quality of dp-promise, DPDM (FID), DP-MEPF, DPDM (Pub), and DP-Diffusion on CelebA and CIFAR-10 under fixed privacy budgets $\varepsilon = \{1, 5, 10\}$. We set $\delta = 10^{-6}$ and $\delta = 10^{-5}$ for CelebA and CIFAR-10, respectively.

The visualization results of these approaches on CelebA and CIFAR-10 are presented in Figure 5. We can observe that dp-promise is able to generate realistic samples under

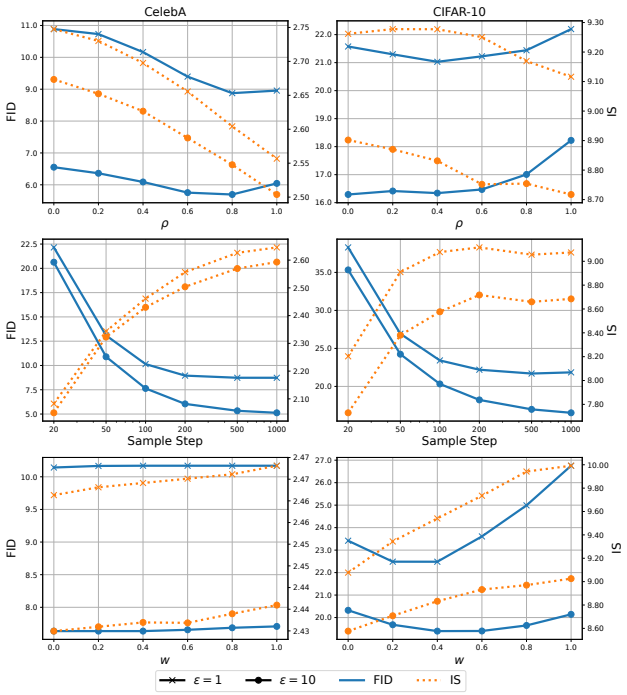


Figure 6: The influence of sampling hyper-parameters on CelebA (left column) and CIFAR-10 (right column). Three rows represent randomness ρ, sample step T', and guidance scale w, respectively.

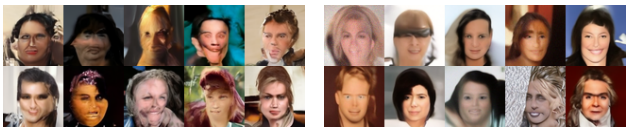
a standard privacy budget (e.g., $\varepsilon = \{1, 10\}$). We report the sample quality in Table 3. It is shown that dp-promise can generate samples that are close to original images, achieving a lower FID and higher Inception Score compared to the baselines.

Effect of Phase I. To investigate the impact of Phase I on sample quality and downstream utility, we conducted ablation experiments on Phase I. For the experimental settings, we adjusted S to 800 and retained most of the previous experimental settings on MNIST and Fashion-MNIST under $\varepsilon = 10$ and $\delta = 10^{-5}$. Note that the purpose of reducing the value of S here is to increase the ratio of Phase I over all time-steps while increasing the privacy budget of Phase I. As shown in Table 4, Phase I enhances the sample quality and downstream utility. This is attributed to the network in DMs needing to fully learn the reverse process from 1 to T. Therefore, it is necessary to train the network from S to T. However, since most denoising steps are in Phase II, the network cannot solely learn the reverse process through Phase I. It still needs to learn the denoising steps with smaller noise through Phase II while also providing privacy guarantees.

Effect of hyper-parameters for sampling. In this experiment, we investigate the impact of hyper-parameters for sampling on the sample quality of dp-promise. We perform on CelebA and CIFAR-10 under a fixed privacy budget of

Table 5: The FID of synthetic data generated by DPDM and dp-promise on CelebA with a resolution of 64×64 pixels under $\varepsilon = \{1, 5, 10\}$ and $\delta = 10^{-6}$.

Methods	$\varepsilon = 10$		$\varepsilon = 5$		$\varepsilon = 1$	
	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑
DPDM (Pub)	46.5	2.0	50.2	2.1	58.3	2.5
dp-promise (this work)	**25.3**	**2.5**	**26.2**	**2.6**	**29.1**	**2.7**



(a) DPDM (Pub) (b) dp-promise (this work)

Figure 7: Synthetic data generated by DPDM and dp-promise under $\varepsilon = 10$ and $\delta = 10^{-6}$ on CelebA with a resolution of 64×64 pixels.

$\varepsilon = \{1, 10\}$. For randomness ρ, we vary ρ from 0 to 1 while keeping fixed sample step $T' = 200$ and guidance scale $w = 0$. In the first row of Figure 6, we observe that stochastic sampling leads to a lower FID, indicating improved sample quality. However, there is a slight decrease in the Inception Score, suggesting a minor loss in the diversity of the sample. This indicates that stochastic sampling is more robust when dealing with an imprecise neural network in DMs. For sample step T', we vary T' from 20 to 1,000 while keeping fixed $\rho = 1$ and guidance scale $w = 0$. In the second row of Figure 6, the results show that increasing the sample step results in better sample quality with more sampling time. Therefore, we choose $T' = 200$ as the sample step to balance the sample quality and sampling time. Note that DPDM uses a more advanced sampler and considers $T' = 1,000$ as the sample step, consuming a significant amount of time to generate samples. For the guidance scale, we vary w from 0 to 1 while keeping fixed $T' = 200$ and $\rho = 1$. In the third row of Figure 6, we notice that as the guidance scale increases, both FID and Inception Score also improve. This suggests that using a higher guidance scale results in samples with greater diversity. These findings help us understand how different hyper-parameters influence the sample quality of dp-promise and enable us to make choices to balance quality and efficiency in practice.

Higher resolution results. To explore the performance on higher-dimensional datasets, we consider experiments on CelebA with 64×64 resolution under $\varepsilon = \{1, 5, 10\}$. We initiate the pre-training of DMs with 35 million parameters on ImageNet [9] as the public dataset, where each image is resized into 64×64 resolution. The network includes 128 base channels, channel multipliers $(1, 2, 2, 2)$, and attention resolution 16. Pre-training is performed with a learning rate of 2×10^{-4} and 50 epochs at batch size 128. We fine-tune the network with time-step boundary $S = 950$, learning rate $\eta_1 = 3 \times 10^{-4}, \eta_2 = 3 \times 10^{-4}$, batch size $m_1 = 32, m_2 = 4,096$ for $\varepsilon = 10$, batch size $m_1 = 16, m_2 = 4,096$ for $\varepsilon = \{1, 5\}$, the number of iteration $N_1 = 1 \cdot n/m_1, N_2 = 15 \cdot n/m_2$, noise augmentation $K = 4$, and clipping constant $C = 10^{-2}$. In the sampling procedure, we consider unconditional generation with parameters $T' = 200$, $\rho = 1$, and $w = 0$. Since DPDM with public data pre-training is closest to dp-promise in technique and performance, we conduct experiments compared to DPDM (Pub). As shown in Table 5 and Figure 7, dp-promise advances the performance of sample quality. Compared with DPDM, dp-promise generates faces with more fidelity.

7 Conclusion

In this paper, we propose dp-promise, a novel framework to train differentially private DMs. dp-promise contains a two-phase training process that takes advantage of DMs to reduce information loss during private training. Moreover, we provide a rigorous theoretical analysis for dp-promise. The experiments demonstrate a non-trivial improvement over the existing state-of-the-art on typical benchmarks. Furthermore, dp-promise is able to perform under various privacy budgets and on more challenging datasets (e.g., CelebA and CIFAR-10). In summary, this work establishes a connection between DMs and privacy and demonstrates that DMs are a superior choice for differentially private image synthesis.

Comparing the experimental results from higher-dimensional datasets and lower-dimensional datasets, dp-promise shows a performance drop on the higher-dimensional datasets. In the future, we will work on approaches that can provide consistent performance on both higher- and lower-dimensional data.

Acknowledgments

Haichen Wang, Shuchao Pang, and Yihang Rao are financially supported by the National Key R&D Program of China (No.2023YFB2703904) and National Natural Science Foundation of China (No.62206128). Zhigang Lu is financially supported by the JCU Early Career Researcher Grant Scheme. Yongbin Zhou is financially supported by the National Key R&D Program of China (No.2022YFB3103800) and National Natural Science Foundation of China (No.U2336205). Minhui Xue is financially supported by the Australian Research Council (ARC) (No.DP240103068) and CSIRO – National Science Foundation (US) AI Research Collaboration Program. Shuchao Pang is the corresponding author.

Availability

The code of our experiments is available at https://github.com/deabfc/dp-promise.

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.

[2] Alex Bie, Gautam Kamath, and Guojun Zhang. Private GANs, revisited. In *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*, 2022.

[3] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. Deep learning with gaussian differential privacy. *Harvard data science review*, 2020(23):10–1162, 2020.

[4] Zhiqi Bu, Sivakanth Gopi, Janardhan Kulkarni, Yin Tat Lee, Hanwen Shen, and Uthaipon Tantipongpipat. Fast and memory efficient differentially private-sgd via jl projections. *Advances in Neural Information Processing Systems*, 34:19680–19691, 2021.

[5] 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] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023.

[7] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. *Advances in Neural Information Processing Systems*, 33:12673–12684, 2020.

[8] Jia-Wei Chen, Chia-Mu Yu, Ching-Chia Kao, Tzai-Wei Pang, and Chun-Shien Lu. Dpgen: Differentially private generative energy-guided network for natural image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8387–8396, 2022.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[11] Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *Transactions on Machine Learning Research*, 2023.

[12] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):3–37, 2022.

[13] Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 8717–8730. PMLR, 23–29 Jul 2023.

[14] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pages 486–503. Springer, 2006.

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[16] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[17] 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.

[18] 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.

[19] Frederik Harder, Milad Jalali, Danica J Sutherland, and Mijung Park. Pre-trained perceptual features improve differentially private image generation. *Transactions on Machine Learning Research*, 2023.

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[21] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proc. Priv. Enhancing Technol.*, 2019(4):232–249, 2019.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[24] Yuzheng Hu, Fan Wu, Qinbin Li, Yunhui Long, Gonzalo Munilla Garrido, Chang Ge, Bolin Ding, David Forsyth, Bo Li, and Dawn Song. Sok: Privacy-preserving data synthesis. *arXiv preprint arXiv:2307.02106*, 2023.

[25] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

[26] Mostafa Kahla, Si Chen, Hoang Anh Just, and Ruoxi Jia. Label-only model inversion attacks via boundary repulsion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15045–15053, June 2022.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] Oliver Kramer and Oliver Kramer. Scikit-learn. *Machine learning for evolution strategies*, pages 45–53, 2016.

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

[30] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010.

[31] Ninghui Li, Wahbeh Qardaji, and Dong Su. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pages 32–33, 2012.

[32] Seng Pei Liew, Tsubasa Takahashi, and Michihiko Ueno. PEARL: Data synthesis via private embeddings and adversarial reconstruction learning. In *International Conference on Learning Representations*, 2022.

[33] Chih-Hsun Lin, Chia-Yi Hsu, Chia-Mu Yu, Yang Cao, and Chun-Ying Huang. Dpaf: Image synthesis via differentially private aggregation in forward phase. *arXiv preprint arXiv:2304.12185*, 2023.

[34] Bochao Liu, Shiming Ge, Pengju Wang, Liansheng Zhuang, and Tongliang Liu. Learning differentially private probabilistic models for privacy-preserving image generation. *arXiv preprint arXiv:2305.10662*, 2023.

[35] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[36] Yunhui Long, Boxin Wang, Zhuolin Yang, Bhavya Kailkhura, Aston Zhang, Carl Gunter, and Bo Li. G-pate: Scalable differentially private data generator via private aggregation of teacher discriminators. *Advances in Neural Information Processing Systems*, 34:2965–2977, 2021.

[37] Saiyue Lyu, Margarita Vinaroz, Michael F Liu, and Mijung Park. Differentially private latent diffusion models. *arXiv preprint arXiv:2305.15759*, 2023.

[38] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[39] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pages 866–882. IEEE, 2021.

[40] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[41] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with PATE. In *International Conference on Learning Representations*, 2018.

[42] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):9312–9321, 2021.

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen,

Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[44] 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.

[45] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[46] Tom Sander, Pierre Stock, and Alexandre Sablayrolles. Tan without a burn: Scaling laws of dp-sgd. In *International Conference on Machine Learning*, pages 29937–29949. PMLR, 2023.

[47] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[49] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[50] 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.

[51] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[52] Margarita Vinaroz, Mohammad-Amin Charusaie, Frederik Harder, Kamil Adamczewski, and Mi Jung Park. Hermite polynomial features for private data generation. In *International Conference on Machine Learning*, pages 22300–22324. PMLR, 2022.

[53] Boxin Wang, Fan Wu, Yunhui Long, Luka Rimanic, Ce Zhang, and Bo Li. Datalens: Scalable privacy preserving training via gradient compression and aggregation. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2146–2168, 2021.

[54] Jiaqi Wang, Roei Schuster, Ilia Shumailov, David Lie, and Nicolas Papernot. In differential privacy, there is truth: on vote-histogram leakage in ensemble private learning. *Advances in Neural Information Processing Systems*, 35:29026–29037, 2022.

[55] 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.

[56] Jianxin Wei, Ergute Bao, Xiaokui Xiao, and Yin Yang. Dpis: An enhanced mechanism for differentially private sgd with importance sampling. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2885–2899, 2022.

[57] Ruihan Wu, Chuan Guo, and Kamalika Chaudhuri. Large-scale public data improves differentially private image generation quality. *arXiv preprint arXiv:2309.00008*, 2023.

[58] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[59] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

[60] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[61] 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.

[62] Lefeng Zhang, Tianqing Zhu, Ping Xiong, Wanlei Zhou, and S Yu Philip. A robust game-theoretical federated learning framework with joint differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3333–3346, 2022.

[63] Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Zhiwei Steven Wu, and Jinfeng Yi. Understanding clipping for federated learning: Convergence and client-level differential privacy. In *International Conference on Machine Learning, ICML 2022*, 2022.

[64] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261, 2020.

[65] Yuqing Zhu and Yu-Xiang Wang. Poission subsampled rényi differential privacy. In *International Conference on Machine Learning*, pages 7634–7642. PMLR, 2019.