Private Gradient Estimation is Useful for Generative Modeling

Bochao Liu
Institute of Information Engineering, Chinese Academy of Sciences
Sch. of Cyb. Sec, UCAS
Beijing, China
liubochao@iie.ac.cn

Pengju Wang
Institute of Information Engineering, Chinese Academy of Sciences
Sch. of Cyb. Sec, UCAS
Beijing, China
wangpengju@iie.ac.cn

Weijia Guo
Institute of Information Engineering, Chinese Academy of Sciences
Sch. of Cyb. Sec, UCAS
Beijing, China
guoweijia@iie.ac.cn

Yong Li
Institute of Information Engineering, Chinese Academy of Sciences
Sch. of Cyb. Sec, UCAS
Beijing, China
liyong@iie.ac.cn

Liansheng Zhuang
Sch. of Cyb. Sec, USTC
Anhui, China
lszhuang@ustc.edu.cn

Weiping Wang
Institute of Information Engineering, Chinese Academy of Sciences
Sch. of Cyb. Sec, UCAS
Beijing, China
wangweiping@iie.ac.cn

Shiming Ge*
Institute of Information Engineering, Chinese Academy of Sciences
Sch. of Cyb. Sec, UCAS
Beijing, China
geshiming@iie.ac.cn

Abstract

While generative models have proved successful in many domains, they may pose a privacy leakage risk in practical deployment. To address this issue, differentially private generative model learning has emerged as a solution to train private generative models for different downstream tasks. However, existing private generative modeling approaches face significant challenges in generating high-dimensional data due to the inherent complexity involved in modeling such data. In this work, we present a new private generative modeling approach where samples are generated via Hamiltonian dynamics with gradients of the private dataset estimated by a well-trained network. In the approach, we achieve differential privacy by perturbing the projection vectors in the estimation of gradients with sliced score matching. In addition, we enhance the reconstruction ability of the model by incorporating a residual enhancement module during the score matching. For sampling, we perform Hamiltonian dynamics with gradients estimated by the well-trained network, allowing the sampled data close to the private dataset's manifold step by step. In this way, our model is able to generate data with a resolution of 256×256. Extensive experiments and analysis clearly demonstrate the effectiveness and rationality of the proposed approach.

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

MM '24, October 28-November 1, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0686-8/24/10
https://doi.org/10.1145/3664647.3681713

CCS Concepts

• **Security and privacy** → *Privacy protections*; • **Computing approachologies** → *Neural networks.*

Keywords

Generative models, differential privacy, gradient estimation

ACM Reference Format:
Bochao Liu, Pengju Wang, Weijia Guo, Yong Li, Liansheng Zhuang, Weiping Wang, and Shiming Ge. 2024. Private Gradient Estimation is Useful for Generative Modeling. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24), October 28-November 1, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3664647.3681713

1 Introduction

Generative models have become indispensable tools across a broad spectrum of machine learning applications, such as image generation [4, 19, 21, 35, 36], text-to-image generator learning [2, 44] and imitation learning [27]. However, according to previous works [17, 58], synthetic data generated by these models could lead to data privacy leakage, as shown in Fig. 1. This issue has attracted significant research interest in developing approaches to protect privacy without reducing the usefulness of the generated data. The challenge is to find a proper balance between privacy and utility.

Differentially private (DP) [13, 14] generative modeling is an intuitive idea for addressing the challenge of privacy leakage, which trains DP generative models for privacy-preserving data generation. Many works [9, 20, 49, 56] adopt generative adversarial networks (GANs) [24] as the underlying generation backbone and incorporate the differential privacy into the training process, thereby bounding the privacy budget of the resulting generator. However,

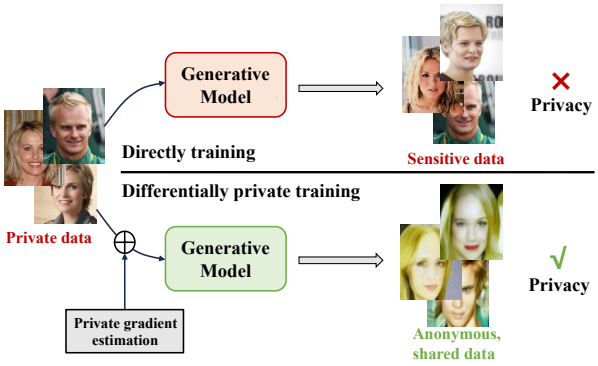

Figure 1: Synthetic data generated by the generative model trained with private data directly may contain sensitive information. To address that, we achieve differentially private learning by private gradient estimation. Synthetic data generated by this generative model can be used for different downstream tasks with privacy protection.

these GAN-based approaches rely on the assumption that the generator can generate the entire real records space to bootstrap the training process and are difficult to converge. Recently, with the potential of diffusion models [28] discovered, many works [10, 12, 39] have begun to explore how to train a privacy-preserving diffusion model. However, the extensive number of queries during training significantly compromises privacy. Consequently, these approaches necessitate thorough pre-training to minimize queries to private data. These challenges culminate in the inability to produce high-resolution images while preserving privacy.

Recent works in the field of generative modeling have underscored the promising capabilities of Energy-Based Models (EBMs) [33] for data generation. EBMs have been found to offer greater stability compared to GANs and require fewer queries to converge in comparison to diffusion models [46, 47]. This observation suggests that EBMs could serve as a solution to the challenges associated with generating high-dimensional data. Additionally, we have discovered that training EBMs with sliced score matching [45] effectively integrates with the randomized response (RR) [52] mechanism, which enables the achievement of differential privacy.

In this work, we propose the **P**rivate **G**radient **E**stimation (PGE) approach that learns to train a DP model to estimate the score of the private data, as shown in Fig. 1, and synthesize privacy-preserving images for downstream tasks. Instead of directly generating images, we train a network to estimate the gradient of logarithmic data density. Inspired by [26], we introduce a residual enhancement module that incorporates masked vectors, obtained by encoding $\boldsymbol{x}$ with a pre-trained VQGAN [15], into the features extracted by the middle layer of the network q_θ to improve its reconstruction ability, as shown in Fig. 2. We adopt sliced score matching to train the network and achieve differential privacy by perturbing the projection vectors with RR. For sampling, as shown in Fig. 3, we design a Markov Chain Monte Carlo (MCMC) sampling approach based on Hamiltonian dynamics, which is more accurate than the commonly used sampling approach based on Langevin dynamics. In section 4 and supplementary material, we provide privacy and convergence analysis to support the effectiveness of our PGE further.

Our PGE approach effectively balances privacy and image quality through four key components. First, it replaces GANs with EBMs to better map sensitive data distributions, improving training stability. Second, we enhance differential privacy by using RR for perturbing proj ection vectors, which reduces randomness and boosts efficiency compared to traditional noisy addition approachs. Third, a residual enhancement module strengthens the network's ability to generate high-fidelity images. Finally, we use Hamiltonian dynamics-based MCMC sampling for more accurate image synthesis. Together, these elements create a solution that ensures both data privacy and high-quality image generation.

Our paper makes several key contributions as follows: (1) we propose the PGE approach, a differentially private generative modeling approach that effectively captures the distribution of private data while preserving valuable information. By leveraging Hamiltonian MCMC sampling, PGE can generate high-resolution images up to 256x256 with exceptional visual quality and data utility; (2) we introduce a residual enhancement module that can be flexibly applied to enhance the reconstruction capabilities of other generative models; (3) we conduct a comprehensive analysis of the privacy and convergence properties of PGE to validate its rationality and effectiveness; (4) experimental results demonstrate that, compared to other existing differentially private generative approaches, PGE significantly improves both the visual quality and data utility of the generated images.

2 Related Works

Differentially private learning. Differentially private learning aims to ensure the training model is differentially private regarding the private data. Existing approaches are typically based on differentially private stochastic gradient descent (DPSGD) [1, 6, 9, 56], which clipped and added noise to the gradients during the training process, and private aggregation of teacher ensembles (PATE) [38, 43, 49], which used semi-supervised learning to transfer the knowledge of the teacher ensemble to the student by a noisy aggregation. Recent works [10, 23, 40] applied randomized response (RR) [52] to the deep learning to achieve differentially private training. Despite significant progress in balancing data privacy and model performance, existing works are still far from optimal in the generative tasks. This is mainly because existing works apply training approachs for discriminative tasks directly to generative tasks. In contrast, we combine sliced score matching and randomized response well to realize differentially private generative modeling.

Generative model learning. With the development of generative techniques, recent works began to train generative models to generate data for downstream tasks. Recent works are typically based on GANs [24] and DDPM [28]. GAN-based approaches [11, 42, 57] are dedicated to improving training stability while improving the quality of the generated images. DDPM-based approaches [34, 41, 46] are committed to improving image generation quality while increasing generation speed. However, the instability of GANs and the high number of queries of DDPM make them difficult to generate high-resolution images under private training. Recently, some works [47, 55] have applied EBMs to generative tasks. It is more

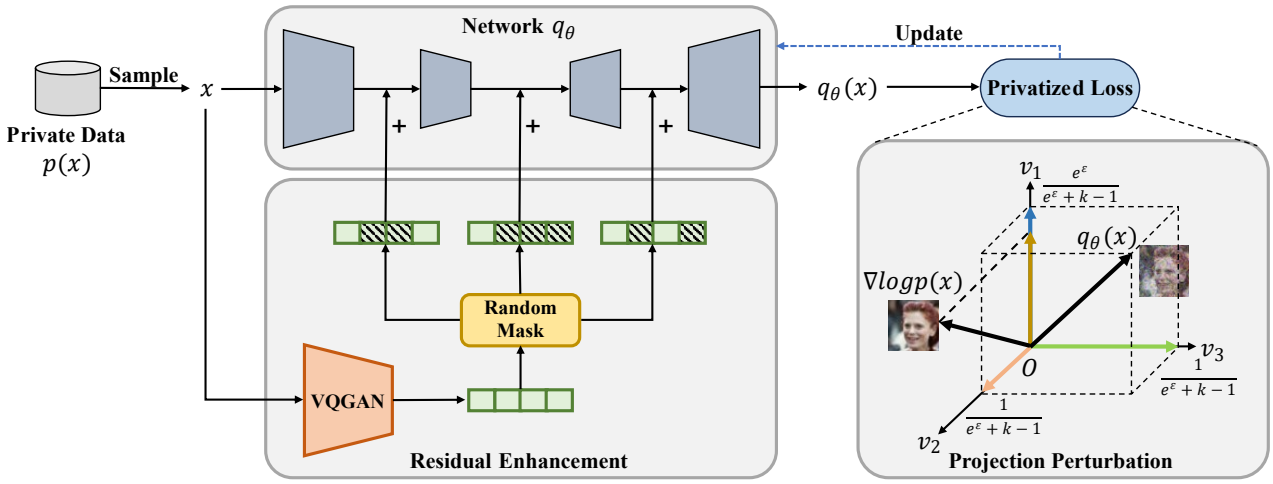

Figure 2: Overview of our PGE. We first sample some images x from the private data distribution $p(x)$. These images are then fed into the network q_θ for prediction. Concurrently, we encode these images using a pre-trained VQGAN and incorporate the masked version into the features extracted by the middle layer of q_θ. This enhances the image reconstruction capability of q_θ. Following the prediction by q_θ, both $q_\theta(x)$ and $p(x)$ are projected for dimensionality reduction. During this process, we perturb their projection vectors by RR to achieve DP. Specifically, $\nabla \log p(x)$ is projected onto the v_1 direction, while RR projects $q_\theta(x)$ onto the v_1 direction with a probability of $e^\varepsilon/(e^\varepsilon+k-1)$, and onto the other direction with a probability of $1/(e^\varepsilon+k-1)$. Here, k refers to the number of projection vectors. Finally, the network q_θ is updated by computing the loss between the predicted distribution $q_\theta(x)$ and the original distribution $p(x)$.

stable than GANs and does not require as many queries as DDPM, which makes it more suitable for private generative modeling.

Feature reconstruction. Feature reconstruction is used in many domains and serves an important role. Masked autoencoder (MAE) [26] have emerged as a cornerstone technique that significantly enhances unsupervised feature learning by reconstructing images from masked inputs, showcasing improved efficiency in various image processing tasks. Following this, [16, 51] extended MAE to video processing, demonstrating reconstruction and feature extraction can be employed to enhance the temporal consistency of video frames. [5, 50] refined MAE applications in facial recognition, leading to advancements in both recognition accuracy and image quality. Face super-resolution [18, 29] can also be regarded as a form of feature reconstruction, where the model learns the mapping from low-dimensional data to high-dimensional data, reconstructing the missing information. Inspired by these works, we design a residual enhancement module to enhance both image quality and robustness of the model.

3 Preliminaries

Energy-based models (EBMs). EBMs capture dependencies by associating a probability density function to each configuration of the given variables. Given a known data distribution $T(\boldsymbol{x})$, we aim to fit it with a probabilistic model $E(\theta;\boldsymbol{x}) = \exp(-H(\theta;\boldsymbol{x}))/Z_\theta$, where $H(\theta;x)$ is an energy function with parameter θ. As $E(\theta;\cdot)$ represents a probability distribution, it needs to be divided by a normalizing constant $Z_\theta = \int \exp(-H(\theta;\boldsymbol{x}))d\boldsymbol{x}$. Z_θ is difficult to calculate explicitly, but as it happens, the image generation process in our paper only requires the gradient of logarithmic data density $\nabla \log E(\theta;\cdot)$, which eliminates the need to compute it in our training. Notably, it is easy to extend to the multi-dimensional case as long as multiple variables are distributed independently of each other.

Differential privacy (DP). DP bounds the change in output distribution caused by a small input difference for a randomized mechanism. It can be described as follows: A randomized mechanism $\mathcal{R}$ with domain $\mathbb{N}^{|x|}$ and range $\mathcal{A}$ is (ε,δ)-differential privacy, if for any subset of outputs $O \subseteq \mathcal{A}$ and any adjacent datasets D and D' :

$$Pr[\mathcal{R}(D) \in O] \leq e^\varepsilon \cdot Pr\left[\mathcal{R}\left(D'\right) \in O\right] + \delta, \tag{1}$$

where adjacent datasets D and D' differ from each other with only one training example. ε is the privacy budget, where the smaller is better, and δ is the failure probability of the mechanism $\mathcal{R}$. In our case, the randomized response mechanism enables the EBMs to satisfy $(\varepsilon, 0)$-DP (or ε-DP). Notably, DP is featured by post-processing theorem and parallel composition theorem. The former could be described as: If $\mathcal{R}$ satisfies (ε,δ)-DP, $\mathcal{R}\circ\mathcal{H}$ will satisfy (ε,δ)-DP for any function $\mathcal{H}$ with $\circ$ denoting the composition operator. And the latter could be described as: If each randomized mechanism $\mathcal{R}_i$ in $\{\mathcal{R}_i\}_{i=1}^n$ satisfies (ε,δ)-DP, then for any division of a dataset $D = \{D_i\}_{i=1}^n$, the sequence of outputs $\{\mathcal{R}_i(D_i)\}_{i=1}^n$ satisfies (ε,δ)-DP regarding the dataset D.

4 Approach

This section outlines our PGE approach, discussing three key aspects. Firstly, we introduce how to fit a data distribution with an EBM privately and how to train a releasable network in practice. Secondly, we describe how to sample to obtain privacy-preserving images. Lastly, we theoretically prove that our PGE can guarantee privacy and convergence.

4.1 Private Gradient Estimation

Physically speaking, we assume that the energy of the private data system is $T(\boldsymbol{x}, \boldsymbol{c}) = U(\boldsymbol{x}) + K(\boldsymbol{c})$, where $\boldsymbol{x}$ represents the position and $\boldsymbol{c}$ represents the velocity. So there is:

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{c}) &\propto \exp\left(-T(\boldsymbol{x}, \boldsymbol{c})\right) \\ &= \exp(-U(\boldsymbol{x}))\exp(-K(\boldsymbol{c})) \\ &\propto p(\boldsymbol{x})p(\boldsymbol{c}), \end{aligned} \tag{2}$$

where $p(\boldsymbol{x})$ and $p(\boldsymbol{c})$ are canonical distributions of position $\boldsymbol{x}$ and velocity $\boldsymbol{c}$, and both are independently distributed. We find that the joint distribution can be sampled using the distributions of the random variables $\boldsymbol{x}$ and $\boldsymbol{c}$. To simplify the calculation, we assume that the distribution of the velocity $\boldsymbol{c}$ is known and that the kinetic energy has:

$$K(\boldsymbol{c}) = -\log p(\boldsymbol{c}) \propto \frac{\boldsymbol{c}^T\boldsymbol{c}}{2}. \tag{3}$$

Similarly, the potential energy function can be expressed as $U(\boldsymbol{x}) = -\log p(\boldsymbol{x})$.

Manifold estimation with an EBM. We use an EBM $E(\theta; \boldsymbol{x}) = \exp(H(\theta; \boldsymbol{x}))/Z_\theta$ to estimate the canonical distribution of energy function $p(\boldsymbol{x})$. As mentioned in the preliminaries section, we only need to estimate the gradient of logarithmic data density $\nabla \log p(\boldsymbol{x})$, so we build the loss function with Fisher divergence as follows:

$$\begin{aligned} &\mathcal{D}_F\left(p(\boldsymbol{x}) \| E(\theta; \boldsymbol{x})\right) \\ &= \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\frac{1}{2}\left\|\nabla \log p(\boldsymbol{x}) - \nabla \log E(\theta; \boldsymbol{x})\right\|^2\right] \\ &= \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\frac{1}{2}\left\|\nabla U(\boldsymbol{x}) - \nabla H(\theta; \boldsymbol{x})\right\|^2\right]. \end{aligned} \tag{4}$$

During the backpropagation process, it is necessary to compute the Hessian matrices of both $p(\boldsymbol{x})$ and $E(\theta; \boldsymbol{x})$. However, due to the high dimensionality of these matrices, the computational requirements are significant. To address this issue, we adopt the random projection approach proposed in [45] to reduce the dimensionality. First, we sample a projection vector $\boldsymbol{v}$ from a standard Gaussian distribution. Then, we project the gradients of $U(\boldsymbol{x})$ and $H(\theta; \boldsymbol{x})$ onto the direction of the projection vector $\boldsymbol{v}$. Finally, compute the loss function as follows:

$$\mathcal{L}\left(\theta; \boldsymbol{x}, \boldsymbol{v}\right) = \quad \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\frac{1}{2}\left\|\boldsymbol{v}^\top \nabla U(\boldsymbol{x}) - \boldsymbol{v}^\top \nabla H(\theta; \boldsymbol{x})\right\|^2\right]. \tag{5}$$

Residual enhancement. In practice, we train a network q_θ instead of $\nabla H(\theta; \boldsymbol{x})$ in Eq. (5) to fit $\nabla U(\boldsymbol{x})$ directly. Our approach can be understood as a process of reconstructing the image information, and inspired by masked autoencoder [26], we add a residual enhancement module, as shown in Fig. 2, to improve the reconstruction ability of the model. Specifically, for a batch of data $\{\boldsymbol{x}_i\}_{i=1}^b$, we enter them into the model q_θ to predict the data manifold. In addition, the data is encoded by a pre-trained VQGAN, and the masked version is incorporated into the features extracted by the middle layer of the model q_θ. This residual enhancement module improves the reconstruction ability of the model [26]. In the sampling process, as shown in Fig. 3, we add noise vectors sampled from a Gaussian distribution to the features instead of masked features encoded by the VQGAN. Adding noise vectors improves the robustness of the model compared to adding nothing, especially when the data are located in low-density regions [47].

Projection perturbation. Training the network q_θ with the gradient of logarithmic data density $\nabla \log p(\boldsymbol{x})$ directly may introduce privacy risks. To address this concern, we perform RR to prevent leakage of private information. It perturbs the training data so that the trained network does not reveal the true distribution of private data during the sampling process. Therefore, we can release the trained network without concern about compromising privacy, as it is difficult for adversaries to tell whether an image is in the training data. Specifically, we aim to privatize the gradient of potential energy function $\nabla \log p(\boldsymbol{x})$ in Eq. (4), which is also equivalent to privatizing $\boldsymbol{v}^\top \nabla U(\boldsymbol{x})$ in Eq. (5). For a batch of data $\boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=1}^b$ and its projection vector $\boldsymbol{v} = \{\boldsymbol{v}_i\}_{i=1}^b$, there is an inherent correspondence between them, so the projection vectors $\boldsymbol{v}_i$ of $\nabla H(\theta; \boldsymbol{x}_i)$ and $\nabla P(\boldsymbol{x}_i)$ are the same without any protection. In our case, we apply a RR mechanism $\mathcal{R}(\cdot)$ to perturb the projection vector of $\nabla U(\boldsymbol{x}_i)$, which protects the private information by making the projection vectors of $\nabla U(\boldsymbol{x})$ and $\nabla H(\theta; \boldsymbol{x})$ not strictly aligned. This perturbation privatizes $\boldsymbol{v}^\top \nabla p(\boldsymbol{x})$, which indirectly randomizes true directions toward perceptually realistic images. The RR mechanism $\mathcal{R}(\cdot)$ can be formulated as follows:

$$\Pr\left[\mathcal{R}\left(\boldsymbol{v}_i\right) = \boldsymbol{v}_o\right] = \begin{cases} \dfrac{e^\varepsilon}{e^\varepsilon + k - 1}, \boldsymbol{v}_o = \boldsymbol{v}_i \\ \dfrac{1}{e^\varepsilon + k - 1}, \boldsymbol{v}_o = \boldsymbol{v}_i' \in \boldsymbol{v}^- \backslash \{\boldsymbol{v}_i\} \end{cases}, \tag{6}$$

where $\boldsymbol{v}^-$ is a subset of $\boldsymbol{v}$, $k = |\boldsymbol{v}^-|$ and $k \geq 2$. We choose the first k projection vectors with the smallest cosine distance from $\boldsymbol{v}_i$ to $\boldsymbol{v}$ to reduce the impact of RR without compromising privacy. Compared to most other approaches with a small failure probability δ, $\mathcal{R}(\cdot)$ achieves pure DP with a failure probability of 0. By this point, the loss function can be expressed as follows:

$$\begin{aligned} &\mathcal{L}\left(\theta; \boldsymbol{x}, \boldsymbol{v}, \mathcal{R}(\cdot)\right) \\ &= \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\frac{1}{2}\left\|\mathcal{R}(\boldsymbol{v}^\top)\nabla U(\boldsymbol{x}) - \boldsymbol{v}^\top \nabla H(\theta; \boldsymbol{x})\right\|^2\right]. \end{aligned} \tag{7}$$

Although we privatize $\nabla \log p(\boldsymbol{x})$, but we cannot obtain $\nabla U(\boldsymbol{x})$ in Eq. (7) directly. Usually, we do not know the exact form of the given data distribution. To address that, we assume that as training proceeds, $q_\theta(\boldsymbol{x})$ converges to $\nabla U(\boldsymbol{x})$ and both satisfy some weak regularization conditions as mentioned in [30]. So combined with [45], our loss function can be formulated as follows:

$$\begin{aligned} &\mathcal{L}\left(\theta; \boldsymbol{x}, \boldsymbol{v}, \mathcal{R}(\cdot)\right) \\ &= \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\mathcal{R}(\boldsymbol{v}^\top)\nabla q_\theta(\boldsymbol{x})\mathcal{R}(\boldsymbol{v}) + \frac{1}{2}\left(\boldsymbol{v}^\top q_\theta(\boldsymbol{x})\right)^2\right]. \end{aligned} \tag{8}$$

In this way, we can estimate the manifold of private data $p(\boldsymbol{x})$ with an EBM $E(\theta; \boldsymbol{x})$. In addition to this, we train the model with the perturbed data manifold estimation to ensure that the model conforms to DP. Overall, PGE can be formally proved ε-DP in Theorem 1. Detailed analysis can be found in supplementary material.

Theorem 1. *Our PGE satisfies ε-DP.*

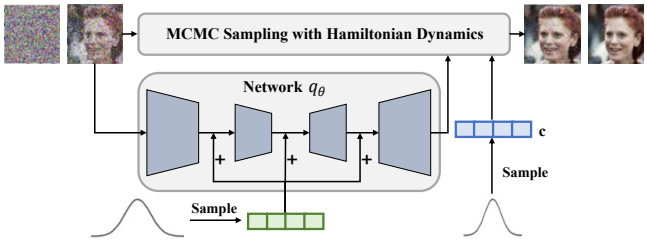

Figure 3: Overview of the sampling process. Given a well-trained network, it can predict the gradients required by Hamiltonian dynamics. After several rounds of sampling, the samples gradually converge from a noisy distribution to the distribution of private data.

4.2 Sampling with Hamiltonian Dynamics

After the training process is completed, we perform MCMC sampling with the leapfrog approach of Hamiltonian dynamics to generate images, as shown in Fig. 3. Following the previous section, for the distribution of private data $p(\boldsymbol{x})$, we have,

$$p(\boldsymbol{x}, \boldsymbol{c}) = p(\boldsymbol{x}) \cdot p(\boldsymbol{c}) \propto \exp(-H(\theta; \boldsymbol{x})) \cdot \exp(-K(\boldsymbol{c})). \tag{9}$$

And the sampling process could be described as follows:

$$\begin{aligned} \boldsymbol{c}(t + \frac{\lambda}{2}) &= \boldsymbol{c}(t) - \frac{\lambda}{2} \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}(t), \boldsymbol{c}(t)); \\ \boldsymbol{x}(t + \lambda) &= \boldsymbol{x}(t) + \lambda \nabla_{\boldsymbol{c}} \log p(\boldsymbol{x}(t), \boldsymbol{c}(t + \frac{\lambda}{2})); \\ \boldsymbol{c}(t + \lambda) &= \boldsymbol{c}(t + \frac{\lambda}{2}) - \frac{\lambda}{2} \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}(t + \lambda), \boldsymbol{c}(t + \frac{\lambda}{2})), \end{aligned} \tag{10}$$

where λ is the step size. According to Eq. (9), we know $-\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}, \boldsymbol{c}) = \nabla_{\boldsymbol{x}} H(\theta; \boldsymbol{x}) = q_\theta(\boldsymbol{x})$ which is the output of the trained network and $-\nabla_{\boldsymbol{c}} \log p(\boldsymbol{x}, \boldsymbol{c}) = \nabla_{\boldsymbol{c}} K(\boldsymbol{c})$. According to Eq. (3), we know $\nabla_{\boldsymbol{c}} K(\boldsymbol{c}) = \alpha \cdot \boldsymbol{c}$. α is the scale factor, which could be combined into the step size λ, so we consider $\nabla_{\boldsymbol{c}} K(\boldsymbol{c}) = \boldsymbol{c}$. After simplification the sampling process is as follows:

$$\begin{aligned} \boldsymbol{c}(t + \frac{\lambda}{2}) &= \boldsymbol{c}(t) + \frac{\lambda}{2} q_\theta(\boldsymbol{x}(t)); \\ \boldsymbol{x}(t + \lambda) &= \boldsymbol{x}(t) - \lambda \boldsymbol{c}(t + \frac{\lambda}{2}); \\ \boldsymbol{c}(t + \lambda) &= \boldsymbol{c}(t + \frac{\lambda}{2}) + \frac{\lambda}{2} q_\theta(\boldsymbol{x}(t + \lambda)), \end{aligned} \tag{11}$$

In this way, the distribution of $\boldsymbol{x}(T)$ will converge infinitely to $p(\boldsymbol{x})$ when $T \to \infty$, in which case $\boldsymbol{x}(T)$ could be considered an exact sample from $p(\boldsymbol{x})$ under some regularity conditions [53]. Moreover, subsequent to executing the described sampling procedure for a predefined number of iterations (N), the Metropolis Criteria are employed to adjudicate the acceptance of the image's most recent state. This decision is governed by the acceptance probability, expressed as $\min(1, q_\theta(\boldsymbol{x}(t + N\lambda))/q_\theta(\boldsymbol{x}(t)))$. Such a mechanism ensures a thorough exploration of the entire distribution of private data, denoted as $p(\boldsymbol{x})$, thereby facilitating the generation of images with enhanced realism. Inspired by the learning rate adjustment technique in machine learning, we adopt the strategy of step size decay to speed up the sampling process. Initially, a larger step size is utilized to expedite the movement towards realistic images. Subsequently, a smaller step size is employed to refine the image details. As an added benefit, because the starting step size is larger, it somewhat also prevents the model from collapsing into certain data-dense regions. Notably, during the sampling process, we no longer use VQGAN to encode the model's input but instead sample directly from a Gaussian distribution to add to the features extracted by the model. On one hand, it speeds up the sampling process compared to using VQGAN encoding, and on the other, it improves the robustness of the model, especially in increasing the diversity of generated data.

4.3 Convergence Analysis

In our convergence analysis, we primarily focus on the model parameters. The foundation of our analysis largely draws upon the methodologies outlined in [3]. We consider a worst-case scenario where the output and input of the RR mechanism differ, resulting in opposite directions of the gradients. Under this condition, applying algorithm $\mathcal{R}(\cdot)$ to the projection vectors effectively becomes equivalent to its application on the gradient. To better express its properties, we rewrite $q_\theta(\cdot)$ as $q(\theta; \cdot)$. Adhering to the same five assumptions in [3], we posit (1) $||\nabla q(\theta; \cdot) - \nabla q(\theta'; \cdot)||_2 \leq \kappa ||\theta - \theta'||_2$; (2) $q(\theta; \cdot) \geq q(\theta'; \cdot) + \nabla q(\theta'; \cdot)^T (\theta - \theta') + \frac{1}{2} c ||\theta - \theta'||_2^2$; (3) $\nabla q(\theta; \cdot)^T \mathbb{E}_{\boldsymbol{x}}[g(\theta; \boldsymbol{x})] \geq \mu ||\nabla q(\theta; \cdot)||_2^2$; (4) $||\mathbb{E}_{\boldsymbol{x}}[g(\theta; \boldsymbol{x})]||_2 \leq \mu_G ||\nabla q(\theta; \cdot)||_2$; (5) $\mathbb{V}_{\boldsymbol{x}}[g(\theta; \boldsymbol{x})] \leq C + \mu_V ||\nabla q(\theta; \cdot)||_2^2$, where θ and θ' are the weights of model q, $\nabla q(\theta; \cdot)$ is the optimal gradient theoretically, $g(\theta, \boldsymbol{x})$ is the gradient we computed, $\mathbb{E}[\cdot]$ is the symbol for mean calculation, $\mathbb{V}[\cdot]$ is the symbol for variance calculation and $\kappa, c, \mu, \mu_G, \mu_V, C$ are non-negative constants. Based on these assumptions, we arrive at the following conclusions,

$$\begin{aligned} &\mathbb{E}[q(\theta_{k+1}; \cdot) - q(\theta^*; \cdot)] + \frac{\gamma^2 \kappa C}{4\tau c} \\ &\leq (2\tau c + 1)(\mathbb{E}[q(\theta_k; \cdot) - q(\theta^*; \cdot)] + \frac{\gamma^2 \kappa C}{4\tau c}), \end{aligned} \tag{12}$$

where $\tau = -\gamma(\frac{2e^\epsilon}{e^\epsilon + k - 1} - 1)\mu + \frac{1}{2}\gamma^2 \kappa(\mu_G^2 + \mu_V)$. When we guarantee that $\tau < 0$, it will converge and the error from the minimum $q(\theta^*; \cdot)$ is $-\frac{\gamma^2 \kappa C}{4\tau c}$. More details can be found in supplementary material.

5 Experiments

To verify the effectiveness of our proposed PGE, we compare it with 11 state-of-the-art approaches and evaluate the data utility and visual quality on four 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

Datasets. We conduct experiments on four image datasets, including MNIST [32], FashionMNIST (FMNIST) [54], CelebA [37] and LSUN [59]. We use the official preprocessed version with the face alignment and resize the images in CelebA to 64×64 and 256×256. CelebA-H and CelebA-G are created based on CelebA with hair color (black/blonde/brown) and gender as the label. For LSUN, we choose the bedroom category and resize the images to 256×256 to evaluate the perceptual scores.

Baselines. We compare our PGE with 14 state-of-the-art approaches, including 7 DPSGD-based approaches (DP-GAN [56], DP-MERF [25],

Table 1: Classification accuracy comparisons with 14 state-of-the-art baselines under different privacy budget ε.

	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 (arXiv'18)	0.4036	0.8011	0.1053	0.6098	0.5330	0.5211	0.3447	0.3920
PATE-GAN (ICLR'19)	0.4168	0.6667	0.4222	0.6218	0.6068	0.6535	0.3789	0.3900
GS-WGAN (NeurIPS'20)	0.1432	0.8075	0.1661	0.6579	0.5901	0.6136	0.4203	0.5225
DP-MERF (AISTATS'21)	0.6367	0.6738	0.5862	0.6162	0.5936	0.6082	0.4413	0.4489
P3GM (ICDE'21)	0.7369	0.7981	0.7223	0.7480	0.5673	0.5884	0.4532	0.4858
G-PATE (NeurIPS'21)	0.5810	0.8092	0.5567	0.6934	0.6702	0.6897	0.4985	0.6217
DataLens (CCS'21)	0.7123	0.8066	0.6478	0.7061	0.7058	0.7287	0.6061	0.6224
DPGEN (CVPR'22)	0.9046	0.9357	0.8283	0.8784	0.6999	0.8835	0.6614	0.8147
DPSH (NeurIPS'21)	N/A	0.8320	N/A	0.7110	N/A	N/A	N/A	N/A
PSG (NeurIPS'22)	0.8090	0.9560	0.7020	0.7770	N/A	N/A	N/A	N/A
DPAC (CVPR'23)	N/A	0.8800	N/A	0.7300	N/A	N/A	N/A	N/A
With pre-training								
DP-DM (TMLR'23)	0.9520	0.9810	0.7940	0.8620	0.7108	0.8586	0.7513	0.8018
DPGU (arXiv'23)	N/A	**0.9860**	N/A	N/A	N/A	N/A	N/A	N/A
DP-LDM (arXiv'23)	0.9590	0.9740	0.7890	0.8514	0.6572	0.8417	0.6851	0.7846
PGE (Ours)	**0.9612**	0.9751	**0.8359**	**0.8934**	**0.7321**	**0.8983**	**0.7153**	**0.8401**

GS-WGAN [9], P3GM [48], PSG [8], DPSH [6], DP-DM [12], DPGU [22], DPAC [7] and DP-LDM [39]), 3 PATE-based approaches (PATE-GAN [31], G-PATE [38] and DataLens [49]) and DPGEN [10] based randomized response. We get the experimental results from their original papers or run their official codes.

Implementations. We choose the same UNet as DP-DM to fit the potential energy of the system. When comparing classifier performance, we choose the same architecture as the other baselines. We initialize the network and the classifier using Kaiming initialization. If not emphasized, we set k to 10 by default and $\boldsymbol{c}$ to sample from a Gaussian distribution by default. The training epoch of network is 10,000 for MNIST, FashionMNIST and 50,000 for CelebA, LSUN. For each dataset, we generate 10,000 samples for classifier learning. The initial value of step size λ is 10^{-5} and the sampling epoch is 1,000. We perform Metropolis Guidelines to decide whether to accept every 100 epochs of sampling.

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

5.2 Experimental Results

Classification accuracy comparisons. In order to demonstrate the effectiveness of our approach, we compare it with 11 state-of-the-art baselines under two privacy budget setting $\varepsilon = 1$ and $\varepsilon = 10$ on MNIST, FMNIST, CelebA-H and CelebA-G. Both our PGE and DPGEN satisfy pure DP, while the other baselines have a failure probability $\delta = 10^{-5}$. We evaluate the classification accuracy of the classifiers trained on the generated data, and the results are summarized in Tab. 1. It is important to note that our approach does not require pre-training with any dataset. Compared to approaches without pre-training, we observe consistent and significant improvements of around 4-6% across different configurations. This improvement is attributed to the fact that most approaches without pre-training rely on GANs and Gaussian mechanism for differentially private generative modeling. In contrast, our approach combines a more stable EBM with RR to achieve a better balance between privacy and data utility. Furthermore, when compared to two DDPM-based approaches with pre-training, our approach consistently achieves optimal results in most settings. This can be attributed to the fact that EBMs converge with fewer queries compared to DDPM, resulting in better performance. These results suggest that our PGE can effectively generate high-resolution images with practical applications.

Table 2: Perceptual scores comparisons with 9 state-of-the-art baselines on CelebA at 64×64 resolution under different privacy budget ε.

Approach	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
PGE (Ours)	10	**2.14**	**12.583**

Perceptual scores comparisons. To further demonstrate the effectiveness of our approach, we evaluate it using two metrics: IS and FID, as mentioned earlier. Since there are no corresponding experimental results for PSG and DP-DM, we compared our approach with the remaining 9 approaches. The results are shown in Tab. 2. Here, a superior IS value signifies enhanced quality of the generated samples, whereas a lower FID score indicates a closer resemblance to authentic images. Our approach outperforms the other baseline approaches by achieving the highest IS of 2.14 and the lowest FID of 12.583, particularly notable under the most stringent privacy budget of 10. This can be attributed to two main factors.

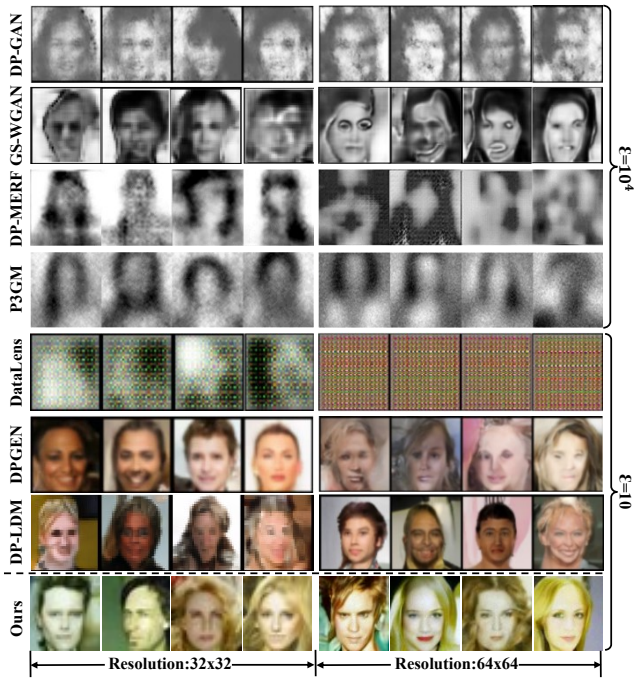

Figure 4: Visualization results of DP-GAN, GS-WGAN, DP-MERF, P3GM, DataLens, DPGEN, DP-LDM and our PGE on CelebA at 32×32 and 64×64 resolutions.

Firstly, we achieve differential privacy by employing RR instead of the Gaussian mechanism, which helps to avoid direct damage to the gradients. Secondly, EBMs exhibit better stability during the training process compared to GANs. These results emphasize the robustness of our approach. Despite the randomized response perturbation, the trained network can still accurately predict the positions of the realistic images.

Visual comparisons of generated data. Furthermore, we provide visual evidence to demonstrate the excellent quality of the generated data by our approach. We compare the visualization results with other baselines in Fig. 4. Even under a high privacy budget condition of $\varepsilon = 10^4$, the grayscale images generated by DP-GAN, GS-WGAN, DP-MERF, and P3GM are still blurry in comparison. Grayscale images have lower dimensionality compared to color images, which makes it relatively easier to balance data quality with privacy protection. The color images generated by DPGEN and DP-LDM exhibit better visual quality than DataLens but lack fleshed-out facial details. In comparison, the images generated by our PGE appear more realistic and have more complete facial details, further confirming the effectiveness of our approach.

Visualization for images with 256 resolution. To further elucidate the capacity of our PGE to generate high-resolution images, we conduct visualizations of images at a resolution of 256, under a setting of $\varepsilon = 20$. The results are presented in Fig. 5, where the generated images serve as empirical evidence to the efficacy of our approach. Notably, despite the augmented complexity inherent to processing images of a 256×256 resolution, our PGE exhibits a robust ability to accurately model the underlying data distribution.

Figure 5: Visualization results of CelebA and LSUN at 256×256 resolution under $\varepsilon = 20$.

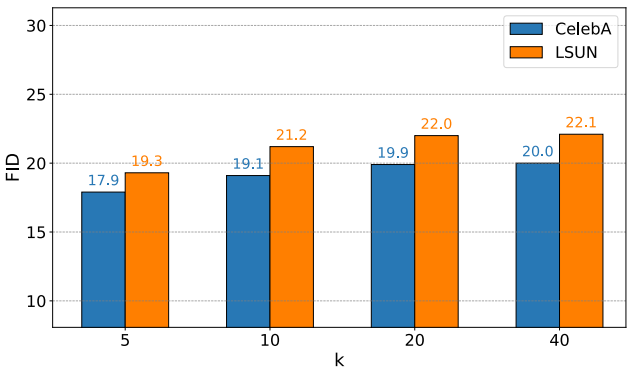

Figure 6: FID score on LSUN and CelebA at 256×256 resolution under different k.

This is a critical observation, as it indicates the preservation of essential visual features even at higher resolutions. Moreover, a detailed comparison with images of lower resolutions (those less than 256) reveals a significant enhancement in the richness of detail within these higher-resolution images. Such improvement is not merely cosmetic but has substantial implications for the utility of the generated data. Specifically, the enhanced detail facilitates the deployment of these images in more demanding downstream tasks, where fine-grained visual information is pivotal. Thus, the results highlight the effectiveness of our approach with respect to the need for high-resolution image generation.

5.3 Ablation Studies

After the promising performance is achieved, we further analyze the impact of each component of our approach, including the hyperparameter k, the step size λ and the initial distribution of kinetic energy $K(\boldsymbol{c})$.

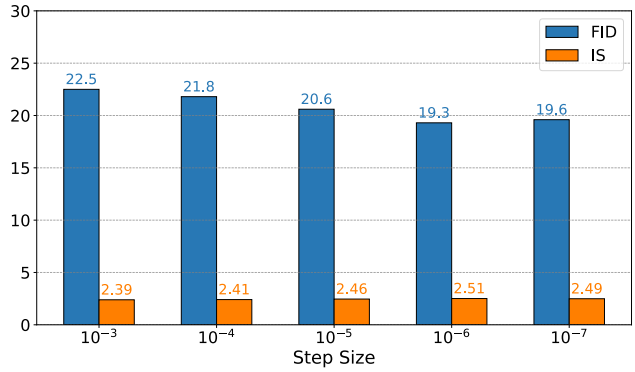

Figure 7: Perceptual scores on LSUN at 256×256 resolution under different step size λ.

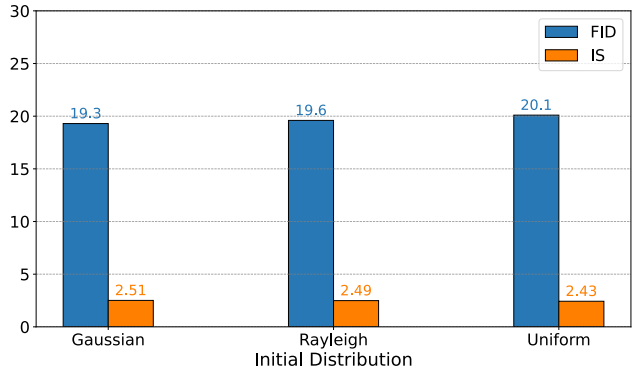

Figure 8: Perceptual scores on LSUN at 256×256 resolution under different kinetic energy distribution $K(\boldsymbol{c})$.

Impact of k. To investigate the impact of the hyperparameter k on the trade-off between privacy and data utility, we compare the perceptual scores obtained when k takes different values under the same privacy budget $\varepsilon = 10$. To make the experimental results more representative, we chose two datasets, CelebA and LSUN, with a resolution of 256×256. The results are presented in Fig. 6. As k increases, the FID score increases, indicating a decrease in the quality of the generated image. This aligns with our expectations. According to Eq. (6), a smaller k value increases the likelihood of $\mathcal{R}(v_i) = v_i$, which in turn enhances the probability that the model accurately captures the underlying data manifold. This implies that the generated data will resemble the distribution of private data more closely. Therefore, as k increases, there is a slight decline in the quality of the generated images.

Impact of step size. To study the effect of the step size λ on the quality of the generated images, we generate images with different step size λ and compare their perceptual scores. The results are shown in Fig. 7. Our findings suggest that decreasing the step size λ weakly improves the quality of the generated images when λ is below $1e-6$. However, at $\lambda = 1e-7$, we observe a slight decrease in the quality of the generated images compared to $\lambda = 1e-6$. Similar to the learning rate in machine learning, a smaller λ leads to finer adjustments per sample, potentially enriching image detail. Nevertheless, excessively small λ values may cause the generation process to converge to local optima, compromising the quality of the resulting images. Therefore, achieving the best balance in choosing $\lambda$is critical to enhancing image detail while ensuring that it doesn't get in the way of other optimization workflows.

Impact of kinetic energy distribution. To explore the effect of the initial distribution of kinetic energy $K(\boldsymbol{c})$ (or $\boldsymbol{c}$) on the data quality. Images are sampled with different initial distribution (Gaussian, Rayleigh, and Uniform) of $\boldsymbol{c}$ and show the results in Fig. 8. It is observed that the images generated with the initial value of $\boldsymbol{c}$ sampled from the Gaussian distribution exhibit the highest quality. The Rayleigh distribution, which is the joint distribution of two independent Gaussian distributions, yielded slightly lower-quality images than the Gaussian distribution. The lowest image quality is observed when the initial values of $\boldsymbol{c}$ are sampled from Uniform distribution, but the difference in quality between the images obtained when the initial value of $\boldsymbol{c}$ was sampled from the three distributions is not very large. The conservation of both kinetic and potential energy in a system not affected by external forces is the main reason for this. Additionally, Eq. (11) reveals that there is an interaction between the two energies, and that the varying distributions only impact the initial energy magnitude of the system.

6 Limitations

There are still some limitations to PGE. We summarize them here. (1) Our use of Hamiltonian dynamics for sampling, unlike GANs that generate images all at once, requires iterative querying, making it significantly slower, especially for high-dimensional images—up to a hundred times slower than GANs. (2) The generated images often lack detailed backgrounds; for instance, CelebA images typically have a solid color background. (3) Our approach struggles with class information extraction compared to classifier-guided diffusion models, as embedding labels directly into images lacks theoretical support. Developing a class-guided sampling approach is a future goal. (4) The model may inadvertently learn dataset biases, which we aim to address through data preprocessing in future work.

7 Conclusion

Releasing private data or trained networks can lead to privacy leakage. To ensure secure deployment, we propose a PGE approach that generates privacy-preserving images for various tasks. By integrating the RR mechanism into the training of EBMs, our approach balances privacy and utility more effectively than other state-of-the-art approaches. Additionally, our MCMC sampling algorithm based on Hamiltonian dynamics enhances realistic data generation. Moreover, we conducted detailed privacy and convergence analyses for our PGE. Notably, PGE satisfies pure DP, eliminating the failure probability present in most other DP generative approachs. Extensive experiments and privacy and convergence analysis are conducted to show the effectiveness and rationality of our approach.

Acknowledgements. This work was supported by grants from the Pioneer R&D Program of Zhejiang Province (2024C01024).

References

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, et al. 2016. Deep learning with differential privacy. In *ACM CCS*. 308–318.

[2] Namhyuk Ahn, Patrick Kwon, Jihye Back, et al. 2023. Interactive Cartoonization with Controllable Perceptual Factors. In *CVPR*. 16827–16835.

[3] Léon Bottou, Frank E Curtis, and Jorge Nocedal. 2018. Optimization methods for large-scale machine learning. *SIAM Rev.* (2018), 223–311.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*.

[5] Zhixi Cai, Shreya Ghosh, Kalin Stefanov, et al. 2023. Marlin: Masked autoencoder for facial video representation learning. In *CVPR*. 1493–1504.

[6] Tianshi Cao, Alex Bie, Arash Vahdat, et al. 2021. Don't generate me: Training differentially private generative models with sinkhorn divergence. In *NeurIPS*. 12480–12492.

[7] Chen Chen, Daochang Liu, Siqi Ma, et al. 2023. Private image generation with dual-purpose auxiliary classifier. In *CVPR*. 20361–20370.

[8] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. 2022. Private Set Generation with Discriminative Information. In *NeurIPS*. 14678–14690.

[9] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. 2020. GS-WGAN: A gradient-sanitized approach for learning differentially private generators. In *NeurIPS*. 12673–12684.

[10] Jia-Wei Chen, Chia-Mu Yu, Ching-Chia Kao, et al. 2022. DPGEN: Differentially private generative energy-guided network for natural image synthesis. In *CVPR*. 8387–8396.

[11] Xi Chen, Yan Duan, Rein Houthooft, et al. 2016. InfoGan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*. 2180–2188.

[12] Tim Dockhorn, Tianshi Cao, Arash Vahdat, et al. 2022. Differentially Private Diffusion Models. *arXiv:2210.09929* (2022).

[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, et al. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*. 265–284.

[14] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* (2014), 211–407.

[15] Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *CVPR*. 12873–12883.

[16] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. 2022. Masked autoencoders as spatiotemporal learners. *NeurIPS* (2022), 35946–35958.

[17] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM CCS*. 1322–1333.

[18] Guangwei Gao, Zixiang Xu, Juncheng Li, et al. 2023. CTCNet: A CNN-transformer cooperation network for face image super-resolution. *IEEE TIP* (2023), 1978–1991.

[19] Shiming Ge, Jia Li, Qiting Ye, and Zhao Luo. 2017. Detecting masked faces in the wild with lle-cnns. In *CVPR*. 2682–2690.

[20] Shiming Ge, Bochao Liu, Pengju Wang, et al. 2022. Learning privacy-preserving student networks via discriminative-generative distillation. *IEEE TIP* 32 (2022), 116–127.

[21] Shiming Ge, Shengwei Zhao, Chenyu Li, and Jia Li. 2018. Low-resolution face recognition in the wild via selective knowledge distillation. *IEEE TIP* (2018), 2051–2062.

[22] Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, et al. 2023. Differentially private diffusion models generate useful synthetic images. *arXiv:2302.13861* (2023).

[23] Badih Ghazi, Noah Golowich, Ravi Kumar, et al. 2021. Deep learning with label differential privacy. In *NeurIPS*. 27131–27145.

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. 2014. Generative adversarial nets. In *NeurIPS*. 2672–2680.

[25] Frederik Harder, Kamil Adamczewski, and Mijung Park. 2021. DP-MERF: Differentially private mean embeddings with random features for practical privacy-preserving data generation. In *ICAIS*. 1819–1827.

[26] Kaiming He, Xinlei Chen, Saining Xie, et al. 2022. Masked autoencoders are scalable vision learners. In *CVPR*. 16000–16009.

[27] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *NeurIPS* (2016), 4572–4580.

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *NeurIPS*. 6840–6851.

[29] Hao Hou, Jun Xu, Yingkun Hou, et al. 2023. Semi-cycled generative adversarial networks for real-world face super-resolution. *IEEE TIP* (2023), 1184–1199.

[30] Aapo Hyvärinen. 2005. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research* (2005), 695–709.

[31] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. 2019. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *ICLR*.

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, et al. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* (1998), 2278–2324.

[33] Yann LeCun, Sumit Chopra, Raia Hadsell, et al. 2006. A Tutorial on Energy-Based Learning. In *Predicting Structured Data*. MIT Press.

[34] Yunchen Li, Zhou Yu, Gaoqi He, et al. 2023. SPD-DDPM: Denoising Diffusion Probabilistic Models in the Symmetric Positive Definite Space. *arXiv:2312.08200* (2023).

[35] Bochao Liu, Pengju Wang, and Shiming Ge. 2024. Learning Differentially Private Diffusion Models via Stochastic Adversarial Distillation. In *ECCV*.

[36] Bochao Liu, Pengju Wang, Shikun Li, et al. 2023. Model conversion via differentially private data-free distillation. In *IJCAI*.

[37] Ziwei Liu, Ping Luo, Xiaogang Wang, et al. 2015. Deep learning face attributes in the wild. In *ICCV*. 3730–3738.

[38] Yunhui Long, Boxin Wang, Zhuolin Yang, and others. 2021. G-PATE: Scalable Differentially Private Data Generator via Private Aggregation of Teacher Discriminators. In *NeurIPS*. 2965–2977.

[39] Saiyue Lyu, Margarita Vinaroz, Michael F. Liu, et al. 2023. Differentially Private Latent Diffusion Models. *arXiv:2305.15759* (2023).

[40] Mani Malek Esmaeili, Ilya Mironov, Karthik Prasad, et al. 2021. Antipodes of label differential privacy: PATE and ALIBI. In *NeurIPS*. 6934–6945.

[41] Nithin Gopalakrishnan Nair, Kangfu Mei, and Vishal M Patel. 2023. AT-DDPM: Restoring faces degraded by atmospheric turbulence using denoising diffusion probabilistic models. In *WACV*. 3434–3443.

[42] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier GANs. In *ICML*. 2642–2651.

[43] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, et al. 2017. Semi-supervised knowledge transfer for deep learning from private training data. In *ICLR*.

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, et al. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*. 10684–10695.

[45] Yang Song, Sahaj Garg, Jiaxin Shi, et al. 2019. Sliced score matching: A scalable approach to density and score estimation. In *UAI*.

[46] Yang Song, Jascha Soh-Dickstein, Diederik P.Kingma, et al. 2021. Score-based generative modeling through stochastic differential equations. In *ICLR*.

[47] Yang Song and Ermon Stefano. 2019. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*. 11918–11930.

[48] Shun Takagi, Tsubasa Takahashi, Yang Cao, et al. 2021. P3GM: Private high-dimensional data release via privacy preserving phased generative model. In *ICDE*. 169–180.

[49] Boxin Wang, Fan Wu, Yunhui Long, et al. 2021. DataLens: Scalable Privacy Preserving Training via Gradient Compression and Aggregation. In *ACM CCS*. 2146–2168.

[50] Kai Wang, Bo Zhao, Xiangyu Peng, et al. 2022. Facemae: Privacy-preserving face recognition via masked autoencoders. *arXiv preprint arXiv:2205.11090* (2022).

[51] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. 2023. Videomae v2: Scaling video masked autoencoders with dual masking. In *CVPR*. 14549–14560.

[52] Stanley L. Warner. 1965. Randomized response: a survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* (1965), 63–69.

[53] Max Welling and Yee Whye Teh. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *ICML*.

[54] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747* (2017).

[55] Jianwen Xie, Yaxuan Zhu, Jun Li, et al. 2022. A tale of two flows: Cooperative learning of langevin flow and normalizing flow toward energy-based model. In *ICLR*.

[56] Liyang Xie, Kaixiang Lin, Shu Wang, et al. 2018. Differentially private generative adversarial network. *arXiv:1802.06739* (2018).

[57] Liang Xu, Ziyang Song, Dongliang Wang, et al. 2023. ActFormer: A GAN-based transformer towards general action-conditioned 3D human motion generation. In *CVPR*. 2228–2238.

[58] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, et al. 2019. Neural network inversion in adversarial setting via background knowledge alignment. In *ACM CCS*. 225–240.

[59] Fisher Yu, Ari Seff, Yinda Zhang, et al. 2015. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv:1506.03365* (2015).

Supplementary Material

Procedure of PGE

The procedure of our approach is shown in Alg. 1 and Alg. 2.

Alg. 1 is the training procedure of the network. It can be described as the following four steps:

- randomly sample a batch of data : $\{\boldsymbol{x}_i\}_{i=1}^{b}$.
- randomly sample a batch of vectors $\boldsymbol{v} = \{\boldsymbol{v}_i\}_{i=1}^{b}$.
- form $\boldsymbol{v}^-$ and compute $\boldsymbol{v}^r = \mathcal{R}(\boldsymbol{v}|\boldsymbol{v}^-)$.
- calculate the loss function with Eq. (8) and update the network.

We note that the network is trained to predict the direction where the logarithmic data density grows the most. We are more interested in the direction of the predictions than the scale, so we choose cosine distance to select the top-k nearest projection vectors to form $\boldsymbol{v}^-$.

Alg. 2 is the sampling procedure of generated images. Unlike traditional Hamiltonian dynamics, we adjust the step size λ every fixed number of epochs (line. 4 in Alg. 2). This enables faster sampling and does not get trapped in a local optimum. Every N rounds of sampling, we perform an acceptance-rejection strategy to improve the fidelity and diversity of the generated images (line. 10-11 in Alg. 2).

Algorithm 1 Private Gradient Estimation

Input: Private data $\boldsymbol{x}$, training iterations T, loss function $\mathcal{L}(\theta;\cdot)$, DP mechanism $\mathcal{R}(\cdot)$, randomized response parameter k, learning rate γ

1: Initialize θ_0 randomly
2: **for** $t \in [T]$ **do**
3: Sample a batch of data $\{\boldsymbol{x}_i\}_{i=1}^{b}$ from $\boldsymbol{x}$
4: Sample a batch of vectors $\{\boldsymbol{v}_i\}_{i=1}^{b}$ from a Gaussian distribution
5: Calculate loss $\mathcal{L}(\theta;\{\boldsymbol{x}_i\}_{i=1}^{b},\{\boldsymbol{v}_i\}_{i=1}^{b},\mathcal{R}(\cdot))$
6: Update the network $\theta_{t+1} \leftarrow \theta_t + \gamma\nabla\mathcal{L}$
7: **end for**
8: **return** θ_T
9: **Function** $\mathcal{R}(\boldsymbol{v})$
10: Initialize $\boldsymbol{v}_r$ to be an empty set Φ
11: **for** each $\boldsymbol{v}_i$ in $\boldsymbol{v}$ **do**
12: Select top-k nearest vectors to $\boldsymbol{v}_i$ from $\boldsymbol{v}$ to form $\boldsymbol{v}^-$
13: Append $\boldsymbol{v}_i$ to $\boldsymbol{v}_r$ with probability of $\frac{e^{\varepsilon}}{e^{\varepsilon}+k-1}$ and append other elements in $\boldsymbol{v}^-$ with probability of $\frac{1}{e^{\varepsilon}+k-1}$
14: **end for**
15: **return** $\boldsymbol{v}_r$

Proof of Theorem. 1

Recall the core thought of our PGE, we perturb the projection vector of log data density to achieve differential privacy. We aim to protect the log data density $(\boldsymbol{v}^T)^*\boldsymbol{v}^T\nabla\log p(\boldsymbol{x})$, which guides the network learning. $\boldsymbol{v}^*$ represents the inverse matrix of $\boldsymbol{v}$. Given a batch of data $D = \{\boldsymbol{x}_i\}_{i=1}^{b}$ and projection vector $\boldsymbol{v} = \{\boldsymbol{v}_i\}_{i=1}^{b}$, the probability of $\boldsymbol{v}_i^*\boldsymbol{v}_i\nabla\log p(\boldsymbol{x}_i)$ is as follows:

$$Pr[\boldsymbol{v}_i,\boldsymbol{x}_i|\boldsymbol{v},D] = Pr[\boldsymbol{x}_i|D]\cdot Pr[\boldsymbol{v}_i|\boldsymbol{v}]\cdot Pr[\boldsymbol{v}^-|\boldsymbol{v}_i,\boldsymbol{v}]\cdot Pr[\mathcal{R}(\boldsymbol{v}_i)=\boldsymbol{v}_i^r|\boldsymbol{v}^-]. \tag{13}$$

After we achieve differential privacy by performing randomized response, the probability of the resulting output $\boldsymbol{v}_i^*\boldsymbol{v}_i^r\nabla\log p(\boldsymbol{x}_i) = \boldsymbol{v}_i^*\mathcal{R}(\boldsymbol{v}_i)\nabla\log p(\boldsymbol{x}_i)$ is as follows:

$$\begin{aligned} &Pr[\boldsymbol{v}_i,\boldsymbol{x}_i,\boldsymbol{v}_i^r,\boldsymbol{v}^-|\boldsymbol{v},D] = \\ &Pr[\boldsymbol{x}_i|D]\cdot Pr[\boldsymbol{v}_i|\boldsymbol{v}]\cdot Pr[\boldsymbol{v}^-|\boldsymbol{v}_i,\boldsymbol{v}]\cdot Pr[\mathcal{R}(\boldsymbol{v}_i)=\boldsymbol{v}_i^r|\boldsymbol{v}^-]. \end{aligned} \tag{14}$$

Algorithm 2 MCMC Sampling with Hamiltonian Dynamics

Input: Trained network $q_\theta(\cdot)$, kinetic energy $K(\cdot)$, step size λ_0, private data D, sampling iterations M, acceptance-rejection iterations N

1: Initialize $\boldsymbol{x}(0), \boldsymbol{c}(0)$ randomly
2: **for** $m \in [M]$ **do**
3: Initialize $p(m)$ randomly
4: $\lambda = \lambda_0 \cdot (M/m)^2$
5: **for** $n \in [N]$ **do**
6: $\boldsymbol{c}\left(m+(n+\frac{1}{2})\lambda\right) = \boldsymbol{c}(m+n\lambda) - \frac{\lambda}{2}q_\theta(\boldsymbol{x}(m+n\lambda))$
7: $\boldsymbol{x}(m+(n+1)\lambda) = \boldsymbol{x}(m+n\lambda) + \lambda\nabla_{\boldsymbol{c}}K(\boldsymbol{c}\left(m+(n+\frac{1}{2})\lambda\right))$
8: $\boldsymbol{c}(m+(n+1)\lambda) = \boldsymbol{c}\left(m+(n+\frac{1}{2})\lambda\right) - \frac{\lambda}{2}q_\theta(\boldsymbol{x}(m+(n+1)\lambda))$
9: **end for**
10: Calculate the probability $p_a = \min(1, q_\theta(\boldsymbol{x}(m+N\lambda))/q_\theta(\boldsymbol{x}(m)))$
11: $\boldsymbol{x}(m+1) = \boldsymbol{x}(m+N\lambda)$ with probability p_a and $\boldsymbol{x}(m+1) = \boldsymbol{x}(m)$ with $1-p_a$
12: **end for**
13: **return** $\boldsymbol{x}(M)$

This equation is obtained by Bayes' theorem. In our approach, we sample data $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$ uniformly, so that $Pr[\boldsymbol{x}_i|D]$ and $Pr[\boldsymbol{v}_i|\boldsymbol{v}]$ are $1/b$. Given $\boldsymbol{v}$ and $\boldsymbol{v}_i$, we select the top-k vectors from $\boldsymbol{v}$ that are similar to $\boldsymbol{v}_i$ to form $\boldsymbol{v}^-$. This process is fixed, so $Pr[\boldsymbol{v}^-|\boldsymbol{v}_i,\boldsymbol{v}] = 1$. Following the above analysis, we have,

$$Pr[\boldsymbol{x}_i,\boldsymbol{v}_i,\boldsymbol{v}_i^r,\boldsymbol{v}^-|\boldsymbol{v},D] = Pr[\mathcal{R}(\boldsymbol{v}_i)=\boldsymbol{v}_i^r|\boldsymbol{v}^-]\cdot 1/b^2. \tag{15}$$

Given a image $\boldsymbol{x}_i$ and its projection vector $\boldsymbol{v}_i$, we define $\mathcal{M}(\boldsymbol{v}_i,\boldsymbol{u}_i) = \mathcal{R}(\boldsymbol{v}_i)\cdot\mathcal{H}(\boldsymbol{u}_i) = \boldsymbol{v}_i^r\cdot\nabla\log p(\boldsymbol{x}_i)$. In our case, we assume that $\mathcal{R}$ and $\mathcal{H}$ are independent of each other, so $Pr[\mathcal{M}(\cdot)] = Pr[\mathcal{R}(\cdot)]\cdot Pr[\mathcal{H}(\cdot)]$.

Lemma 1. *For any two different training data $\boldsymbol{x}_i,\boldsymbol{x}_j$ and their projection vectors $\boldsymbol{v}_i,\boldsymbol{v}_j$, the mechanism $\mathcal{M}$ satisfies*

$$Pr\left[\mathcal{M}\left(\boldsymbol{v}_i,\boldsymbol{x}_i\right)\in O\right] \le e^{\varepsilon}\cdot Pr\left[\mathcal{M}\left(\boldsymbol{v}_j,\boldsymbol{x}_j\right)\in O\right], \tag{16}$$

where O is a possible output of $\mathcal{M}$.

Proof. *From the definition of randomized response mechanism, we can know the probability that $\mathcal{R}(\cdot)$ takes as input $\boldsymbol{v}_i$ and returns $\boldsymbol{v}_i$ is the largest for $e^{\varepsilon}/(e^{\varepsilon}+k-1)$ and that takes as input $\boldsymbol{v}_j$ and returns $\boldsymbol{v}_i$ is the smallest for $1/(e^{\varepsilon}+k-1)$. We sample $\boldsymbol{x}_i$ uniformly, so $Pr[\mathcal{H}(\boldsymbol{x}_i)] = Pr[\mathcal{H}(\boldsymbol{x}_j)]$. Then we have*

$$\begin{aligned} Pr\left[\mathcal{M}\left(\boldsymbol{v}_i,\boldsymbol{x}_i\right)\in O\right] &= Pr\left[\mathcal{R}(\boldsymbol{v}_i)=\boldsymbol{v}_o\right]\cdot Pr[\mathcal{H}(\boldsymbol{x}_i)] \\ &\le e^{\varepsilon}\cdot Pr\left[\mathcal{R}(\boldsymbol{v}_j)=\boldsymbol{v}_o\right]\cdot Pr[\mathcal{H}(\boldsymbol{x}_i)] \\ &= e^{\varepsilon}\cdot Pr\left[\mathcal{R}(\boldsymbol{v}_j)=\boldsymbol{v}_o\right]\cdot Pr[\mathcal{H}(\boldsymbol{x}_j)] \\ &= e^{\varepsilon}\cdot Pr\left[\mathcal{M}\left(\boldsymbol{v}_j,\boldsymbol{x}_j\right)\in O\right] \end{aligned} \tag{17}$$

□

Lemma 2. *The mechanism $\mathcal{M}$ satisfies ε-DP.*

Proof. *Consider two adjacent datasets $D = \{\boldsymbol{x}_i\}_{i=1}^b, D' = \{\boldsymbol{x}_i'\}_{i=1}^b$ that differ only by one data and their projection vectors $\boldsymbol{v} = \{\boldsymbol{v}_i\}_{i=1}^b, \boldsymbol{v}' = \{\boldsymbol{v}_i'\}_{i=1}^b$. The data are independent of each other so we have*

$$\begin{aligned} &Pr[\mathcal{M}(\boldsymbol{v}, D) \in O] \\ &= Pr\left[\mathcal{M}\left(\boldsymbol{v} \cap \boldsymbol{v}', D \cap D'\right) \in O\right] \cdot Pr\left[\mathcal{M}\left(\boldsymbol{v}_i, \boldsymbol{x}_i\right) \in O\right] \\ &\leq e^{\varepsilon} \cdot Pr\left[\mathcal{M}\left(\boldsymbol{v} \cap \boldsymbol{v}', D \cap D'\right) \in O\right] \cdot Pr\left[\mathcal{M}\left(\boldsymbol{v}_i', \boldsymbol{x}_i'\right) \in O\right] \\ &= e^{\varepsilon} \cdot Pr[\mathcal{M}(\boldsymbol{v}', D') \in O], \end{aligned} \tag{18}$$

where O is a set of possible outputs. From line 2 to line 3 is based on Lemma. 1. □

Our approach trains the network with datasets D and projection vectors $\boldsymbol{v}$. It is necessary to calculate the joint probability to clarify the association of each vector. Next, we prove that our PGE satisfies differential privacy based on Lemma. 2.

Theorem 1. *Our PGE satisfies ε-DP.*

Proof. *Consider two adjacent datasets $D = \{\boldsymbol{x}_i\}_{i=1}^b$ and $D' = \{\boldsymbol{x}_i'\}_{i=1}^b$ and their projection vectors $\boldsymbol{v} = \{\boldsymbol{v}_i\}_{i=1}^b, \boldsymbol{v}' = \{\boldsymbol{v}_i'\}_{i=1}^b$. We define $\mathcal{F}(\boldsymbol{v}_i, \boldsymbol{x}_i) = \boldsymbol{v}_i^* \mathcal{R}(\boldsymbol{v}_i) \nabla \log p(\boldsymbol{x}_i)$, then according to Eq. (2) and Eq. (3), we have*

$$\begin{aligned} Pr[\mathcal{F}(\boldsymbol{v}, D) \in O] &= \prod_i Pr[\boldsymbol{v}_i, \boldsymbol{x}_i, \boldsymbol{v}_i^r, \boldsymbol{v}^- | \boldsymbol{v}, D] \\ &= \prod_i Pr[\mathcal{R}(\boldsymbol{v}_i) = \boldsymbol{v}_i^r | \boldsymbol{v}^-] \cdot 1/b^2 \\ &= \prod_i Pr[\mathcal{R}(\boldsymbol{v}_i) \cdot \mathcal{H}(\boldsymbol{x}_i) \in O | \boldsymbol{v}^-] \cdot 1/b^2 \\ &= Pr[\mathcal{M}(\boldsymbol{v}, D) \in O] \cdot 1/b^2 \\ &\leq e^{\varepsilon} \cdot Pr[\mathcal{M}(\boldsymbol{v}', D') \in O] \cdot 1/b^2 \\ &= e^{\varepsilon} \cdot Pr[\mathcal{F}(\boldsymbol{v}', \mathcal{D}') \in O], \end{aligned} \tag{19}$$

where O is the range of output of $\mathcal{F}$, from line 4 to line 5 is based on Lemma. 2 and from line 5 to line 6 is the inverse derivation of line 1 to line 4. We note that as long as $\mathcal{F}(\boldsymbol{v}, D)$ satisfies differential privacy, the trained probabilistic model and the images generated with it also satisfy differential privacy according to the post-processing property of differential privacy. So our PGE satisfies ε-DP. □

Covergence Analysis

We begin by stating that most of our analysis process is based on [Bottou et al., 2018]. We consider the worst-case scenario: the gradient of the randomized response algorithm when the outputs and inputs are different is exactly the opposite of the gradient when they are the same. In this case, the $\mathcal{R}(\cdot)$ algorithm performing on the label is equivalent to performing on the gradient. To make the analysis easier and more understandable, we rewrite $q_\theta(\cdot)$ as $q(\theta; \cdot)$ and follow five standard assumptions same as [Bottou et al., 2018],

$$\begin{aligned} &(1)\ ||\nabla q(\theta; \cdot) - \nabla q(\theta'; \cdot)||_2 \leq \kappa ||\theta - \theta'||_2; \\ &(2)\ q(\theta; \cdot) \geq q(\theta'; \cdot) + \nabla q(\theta'; \cdot)^T (\theta - \theta') + \frac{1}{2} c ||\theta - \theta'||_2^2 \\ &(3)\ \nabla q(\theta; \cdot)^T \mathbb{E}_{\boldsymbol{x}}[g(\theta; \boldsymbol{x})] \geq \mu ||\nabla q(\theta; \cdot)||_2^2; \\ &(4)\ ||\mathbb{E}_{\boldsymbol{x}}[g(\theta; \cdot)]||_2 \leq \mu_G ||\nabla q(\theta; \cdot)||_2; \\ &(5)\ \mathbb{V}_{\boldsymbol{x}}[g(\theta; \cdot)] \leq C + \mu_V ||\nabla q(\theta; \cdot)||_2^2, \end{aligned} \tag{20}$$

where θ and θ' are the weights of model q, $\nabla q(\theta; \cdot)$ is the true gradient, $g(\theta, \cdot)$ is the gradient we computed, $\mathbb{E}[\cdot]$ is the symbol for mean calculation, $\mathbb{V}[\cdot]$ is the symbol for variance calculation and $\kappa, c, \mu, \mu_G, \mu_V, C$ are non-negative constants.

Lemma 3. *For any two weights θ and θ', the difference of the objective function $q(\theta) - q(\theta')$ is limited by the distance between the weights.*

$$q(\theta; \cdot) \leq q(\theta'; \cdot) + \nabla q(\theta'; \cdot)^T (\theta - \theta') + \frac{1}{2} \kappa ||\theta - \theta'||_2^2. \tag{21}$$

Proof. Consider any path s from θ' to θ, we have

$$\begin{aligned} &q(\theta; \cdot) - q(\theta'; \cdot) \\ &= \int_s \nabla q(x; \cdot)^T dx \\ &= \int_0^1 \frac{\partial q(s(t); \cdot)}{\partial t} dt \\ &= \int_{\theta'}^{\theta} \nabla q(s(t); \cdot) ds(t) \\ &= \int_{\theta'}^{\theta} \nabla q(\theta'; \cdot) ds(t) + \int_{\theta'}^{\theta} [\nabla q(s(t); \cdot) - \nabla q(\theta'; \cdot)] ds(t) \\ &\leq \int_{\theta'}^{\theta} \nabla q(\theta'; \cdot) ds(t) + \int_{\theta'}^{\theta} \kappa ||s(t) - \theta'||_2 ds(t) \\ &= \nabla q(\theta'; \cdot)^T (\theta - \theta') + \frac{1}{2} \kappa ||\theta - \theta'||_2^2. \end{aligned} \tag{22}$$

□

The inequality is based on assumption (1).

Lemma 4. *For any weight θ, the distance between $q(\theta; \cdot)$ and the minimum value $q(\theta^*; \cdot)$ is limited by $\nabla q(\theta; \cdot)$ as follows*

$$q(\theta; \cdot) - q(\theta^*; \cdot) \leq \frac{1}{2c} ||\nabla q(\theta; \cdot)||_2^2. \tag{23}$$

Proof. According to assumption (2), we can regard the right side of the inequality as a quadratic function on θ. When $\theta = \theta' - \frac{1}{c} \nabla q(\theta'; \cdot)$, it takes the minimum value $q(\theta'; \cdot) - \frac{1}{2c} ||\nabla q(\theta'; \cdot)||_2^2$. Substituting it into assumption (2) and letting $\theta = \theta^*$, we can get Lemma 4. □

According to the assumptions before (We consider the worst-case scenario: the gradient of the randomized response algorithm when the outputs and inputs are different is exactly the opposite of the gradient when they are the same.), we consider the update at step k as

$$\theta_{k+1} = \theta_k - \gamma \cdot \mathcal{R}(g(\theta_k, \cdot)), \tag{24}$$

where $\mathcal{R}(g(\theta_k, \cdot))$ will return $g(\theta_k, \cdot)$ with the probability of $e^{\varepsilon}/(e^{\varepsilon} + k - 1)$ and return $-g(\theta_k, \cdot)$ with the probability of $1 - e^{\varepsilon}/(e^{\varepsilon} + k - 1)$.

Based on Lemma 3, we have

$$q(\theta_{k+1}; \cdot) \leq q(\theta_k; \cdot) - \gamma \nabla q(\theta_k; \cdot)^T \mathcal{R}(g(\theta_k, \cdot)) + \frac{1}{2} \kappa \gamma^2 \underbrace{||\mathcal{R}(g(\theta_k, \cdot))||_2^2}_{||g(\theta_k, \cdot)||_2^2}. \tag{25}$$

Table 3: Classification accuracy comparisons on image datasets under small privacy budget ε.

	MNIST			FMNIST		
ε	0.2	0.6	0.8	0.2	0.6	0.8
DataLens	0.2344	0.4201	0.6485	0.2226	0.3863	0.5534
PGE	**0.4702**	**0.7462**	**0.9211**	**0.4582**	**0.6954**	**0.8002**

Table 4: Classification accuracy and entropy comparisons on image datasets under small privacy budget ε.

Acc./Entropy	MNIST	FMNIST
With Res.	**0.9751/3.21**	**0.8934/3.23**
Without Res.	0.9543/3.14	0.8761/3.11

Taking the expectations on both sides gives

$$\mathbb{E}[q(\theta_{k+1};\cdot) - q(\theta_k;\cdot)] \leq -\gamma \nabla q(\theta_k;\cdot)^T \mathbb{E}[\mathcal{R}(g(\theta_k,\cdot))] + \frac{1}{2}\gamma^2\kappa \underbrace{\mathbb{E}[||g(\theta_k,\cdot)||_2^2]}_{||\mathbb{E}[g(\theta_k,\cdot)]||_2^2 + \mathbb{V}[g(\theta_k,\cdot)]}. \tag{26}$$

According to our pre-assumed scenario,

$$\begin{aligned}\mathbb{E}[\mathcal{R}(g(\theta_k,\cdot))] &= \frac{e^\varepsilon}{e^\varepsilon + k - 1} g(\theta_k,\cdot) \\ &\quad + (1 - \frac{e^\varepsilon}{e^\varepsilon + k - 1}) g(\theta_k,\cdot) \\ &= \underbrace{(\frac{2e^\varepsilon}{e^\varepsilon + k - 1} - 1)}_{\zeta} g(\theta_k,\cdot).\end{aligned} \tag{27}$$

Combined with assumptions (3), (4) and (5), we can get

$$\begin{aligned}&\mathbb{E}[q(\theta_{k+1};\cdot) - q(\theta_k;\cdot)] \\ &\leq -\gamma\zeta \nabla q(\theta_k;\cdot)^T \mathbb{E}[g(\theta_k,\cdot)] + \frac{1}{2}\gamma^2\kappa(||\mathbb{E}[g(\theta_k,\cdot)]||_2^2 + \mathbb{V}[g(\theta_k,\cdot)]) \\ &\leq -\gamma\zeta\mu ||\nabla q(\theta;\cdot)||_2^2 + \frac{1}{2}\gamma^2\kappa(C + (\mu_G^2 + \mu_V)||q(\theta_k;\cdot)||_2^2) \\ &= \underbrace{(-\gamma\zeta\mu + \frac{1}{2}\gamma^2\kappa(\mu_G^2 + \mu_V))}_{\tau} ||q(\theta_k;\cdot)||_2^2 + \frac{1}{2}\gamma^2\kappa C.\end{aligned} \tag{28}$$

If the algorithm converges, it takes $-\gamma\mu + \frac{1}{2}\gamma^2\kappa(\mu_G^2 + \mu_V) < 0$. According to Lemma 4, we can further get

$$\begin{aligned}&\mathbb{E}[q(\theta_{k+1};\cdot) - q(\theta^*;\cdot)] - \mathbb{E}[q(\theta_k;\cdot) - q(\theta^*;\cdot)] \\ &\leq \tau||q(\theta_k;\cdot)||_2^2 + \frac{1}{2}\gamma^2\kappa C \\ &\leq 2\tau c\mathbb{E}[q(\theta_k;\cdot) - q(\theta^*;\cdot)] + \frac{1}{2}\gamma^2\kappa C.\end{aligned} \tag{29}$$

Eq. 29 is transformed to obtain

$$\begin{aligned}&\mathbb{E}[q(\theta_{k+1};\cdot) - q(\theta^*;\cdot)] + \frac{\gamma^2\kappa C}{4\tau c} \\ &\leq (2\tau c + 1)(\mathbb{E}[q(\theta_k;\cdot) - q(\theta^*;\cdot)] + \frac{\gamma^2\kappa C}{4\tau c})\end{aligned} \tag{30}$$

We know $\tau < 0$, so $2\tau c + 1 < 1$. The algorithm converges when we guarantee that $\tau < 0$. The error from the minimum $q(\theta^*;\cdot)$ is $-\frac{\gamma^2\kappa C}{4\tau c}$.

Experimental Results under Small ε

We conduct experiments under small ε to verify the effectiveness of our method here. The results are shown in Tab. 3. We can find that even under the condition of small ε, our method still has outstanding performance.

Discussion of Residual Structure

We use a residual structure in our framework. Here, we conduct experiments on two datasets (MNIST and FMNIST) to explore the role of this structure. The results are shown in Tab. 4. We capture the diversity of the generated samples through the metric of entropy. We find that having this structure improves the diversity and data utility of the generated samples.

Extended Visualization Results

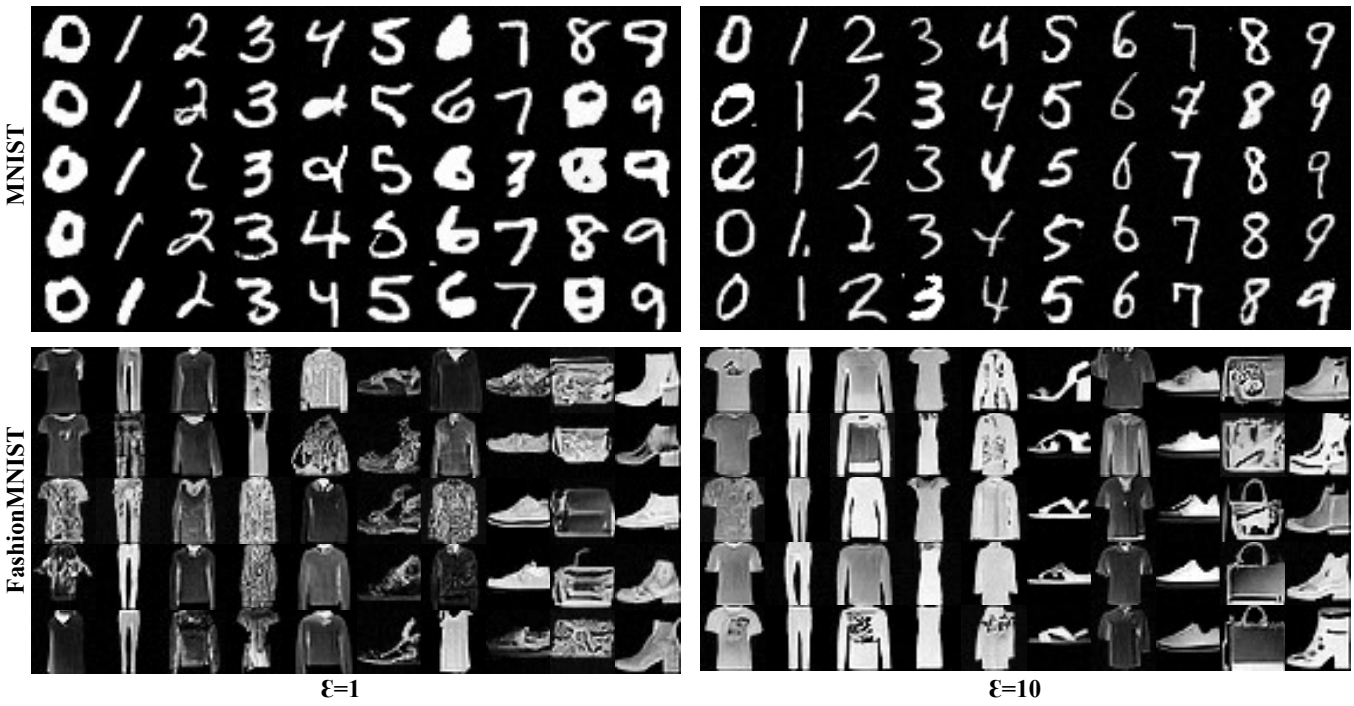

Figure 9: Visualization results of MNIST and FashionMNIST with 28×28 resolution under different privacy budget.

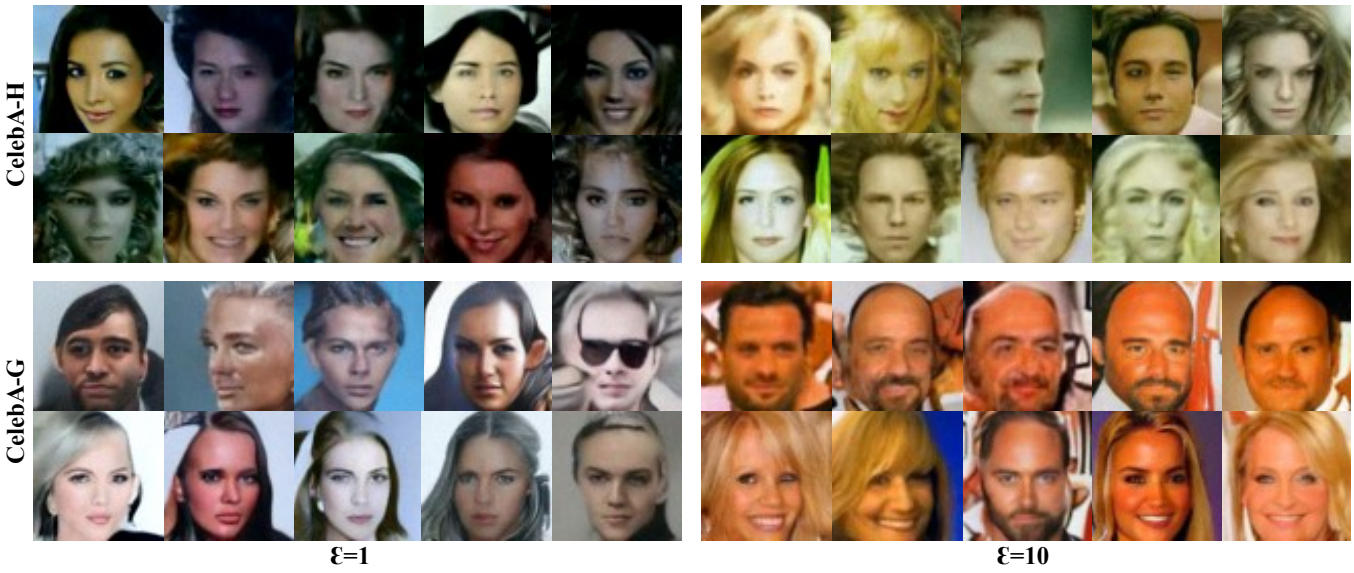

Figure 10: Visualization results of CelebA with 64×64 resolution under different privacy budget.