Learning Differentially Private Probabilistic Models for Privacy-Preserving Image Generation

Bochao Liu[1,2] **Shiming Ge**[1,2*] **Pengju Wang**[1,2] **Liansheng Zhuang**[3]
Tongliang Liu[4]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]School of Data Science, University of Science and Technology of China
[4] Trustworthy Machine Learning Lab, The University of Sydney

Abstract

A number of deep models trained on high-quality and valuable images have been deployed in practical applications, which may pose a leakage risk of data privacy. Learning differentially private generative models can sidestep this challenge through indirect data access. However, such differentially private generative models learned by existing approaches can only generate images with a low-resolution of less than 128×128, hindering the widespread usage of generated images in downstream training. In this work, we propose learning differentially private probabilistic models (DPPM) to generate high-resolution images with differential privacy guarantee. In particular, we first train a model to fit the distribution of the training data and make it satisfy differential privacy by performing a randomized response mechanism during training process. Then we perform Hamiltonian dynamics sampling along with the differentially private movement direction predicted by the trained probabilistic model to obtain the privacy-preserving images. In this way, it is possible to apply these images to different downstream tasks while protecting private information. Notably, compared to other state-of-the-art differentially private generative approaches, our approach can generate images up to 256×256 with remarkable visual quality and data utility. Extensive experiments show the effectiveness of our approach.

1 Introduction

Advanced deep learning approaches have achieved great success [22; 20; 32; 15] mainly due to the accessibility of large-scale datasets. A major concern is that the deployment of these trained models may lead to the risk of privacy leakage due to the sensitive information contained in the training data, such as human faces. Recent works [12; 42] have shown that training data can be easily recovered with little access to the released model. Thus, the importance of privacy protection has become apparent to both the public and academia and it is necessary to explore a feasible solution that protects private information while maintaining the high utility of the dataset.

Differentially private generative model learning provides a compelling solution to such privacy challenge. It is the combination of generative model learning and differential privacy (DP) [11]. Most of the existing approaches typically adopt DPSGD [1] or PATE [31] to equip the generative models with rigorous privacy guarantees and use generative adversarial networks (GANs) [13] as the framework. [41] trained GANs with DPSGD which achieves DP by adding noise to all gradients. Then [5] suggested that only needed to add noisy gradients to the generator's outputs to achieve

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

Preprint. Under review.

differential privacy (DP) [10] and proposed GS-WGAN. [37] further improves upon GS-WGAN by introducing TopAgg gradients aggregation algorithm. But there are still two major drawbacks in differentially private GANs: (1) DP noise destroys the magnitude and direction of the gradients; (2) noisy gradients exacerbate the instability inherent in GANs. These two drawbacks make it highly challenging for these approaches to generate images with resolutions higher than 64×64. Different from the Gaussian mechanism which achieves DP by adding noise to gradients, randomized response (RR) [38] achieves DP by randomizing the outputs of the queries, which avoids direct damage to the gradients. [6] applied RR to perturb training data and trained an energy-guided network instead of a generator to improve the stability of training, but it can only fit noise-perturbed data. Note that this noise is not for achieving DP, it needs to add additional perturbation which could seriously affect the utility of generated data. And this approach is not recognized by [9].

In the meantime, recent researches have found that probabilistic models can also be used for data generation and have better stability than GANs [35; 34]. So in this work, we aim to explore the potential of probabilistic models and DP to achieve privacy-preserving data generation that can be used for different downstream tasks with high utility. Specifically, we generate images by performing Markov Chain Monte Carlo (MCMC) sampling with Hamiltonian dynamics [19; 27], wherein the only unknown movement directions are predicted by a differentially private energy-based model (EBM) [23]. As shown in Fig. 1, the whole process is divided into two parts. In the first step, we train a neural network (NN), which is used to predict the movement directions in MCMC sampling, to fit the distribution of private data. We hope to generate high-resolution images. As we all know, it is difficult to calculate the Hessian matrix of high-dimensional vectors, so we use random projection to reduce the dimension. Notably, the projection vectors of NN prediction and the private data distribution are the same without protection. We perturb the projection vector by performing RR, which randomizes the true directions toward perceptually realistic images indirectly, to achieve DP. In the second step, we perform MCMC sampling with Hamiltonian dynamics to generate images. To simplify the sampling process, we assume that the kinetic energy follows a known distribution, such as Gaussian distribution, and predict the score of potential energy with trained NN. Note that the outputs of the trained NN are not images, but rather directions of travel when images are sampled. In this way, we could generate perceptually realistic images while preserving the private information.

In summary, our DPPM can effectively generate privacy-preserving data with high utility benefits from three points. First, we improve the stability of the training and image generation process by training a probabilistic model instead of GANs to fit the distribution of private data. Second, we avoid noise damage to the gradients by applying the RR mechanism to achieve DP. Third, we increase the probability that the outputs are projected in the same direction by the top-k selection without affecting privacy protection.

We summarize our main contributions as follows, (1) we propose learning differentially private probabilistic models approach DPPM that could fit the private data distribution well while protecting valuable information. Combined with Hamiltonian MCMC sampling, it can generate images up to 256×256 resolution. Notably, these generated images have remarkable visual quality and data utility while protecting the sensitive information of private data; (2) we propose a differentially private score matching approach that achieves pure DP by applying RR to the projection vectors; (3) experimental results demonstrate that, in comparison to other state-of-the-art differentially private generative approaches, our approach significantly improves the visual quality and data utility of generated images.

2 Related Works

The approach we proposed in this paper aims to train a differentially private model that can assist in generating privacy-preserving images. Therefore, we briefly review the related works for two inspects, including differentially private learning and generative model learning.

Differentially private learning. Differentially private learning aims to ensure the training model is differentially private regarding the private data. [1] proposed DPSGD training framework which achieved DP by clipping and adding noise to the gradients of all parameters during the training process. However, the model performance degrades severely with strong privacy requirements. [31] later proposed PATE framework which used semi-supervised learning to transfer the knowledge of the teacher ensemble to the student by using a noisy aggregation. Based on these two frameworks,

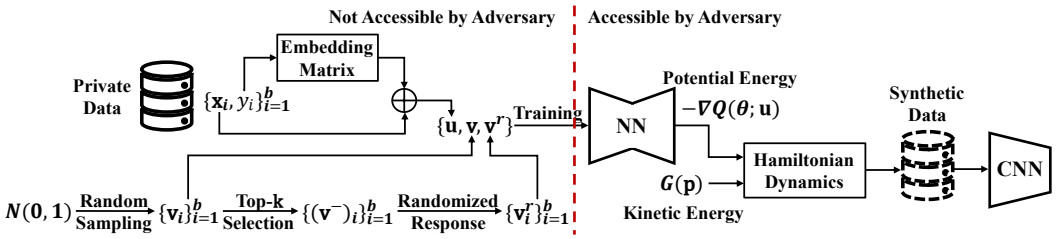


Figure 1: Overview of our DPPM. First we embed the labels into the images through an embedding matrix to get $\mathbf{u}$; Then we sample the projection vectors $\mathbf{v}$ from a standard Gaussian distribution. DP is achieved by perturbing $\mathbf{v}$ with randomized response to get $\mathbf{v}^r$. Top-k selection is used to reduce the impact of randomized response without affecting the privacy budget. $(\mathbf{v}^-)_i$ is the set consisting of the k elements in $\mathbf{v}$ that are closest to $\mathbf{v}_i$; Finally we use triples $\{\mathbf{u}, \mathbf{v}, \mathbf{v}^r\}$ to train the NN (RefineNet [24]). The trained NN is released for predicting the potential energy, and in combination with the known kinetic energy $G(\mathbf{p})$, users could generate images for downstream tasks via Hamiltonian dynamics. In our case, the downstream task is to train a CNN classifier.

some recent works want to train differentially private generators to generate data that has a similar distribution to private data while preserving privacy. [41] applied DPSGD to the training process of GANs to obtain a differentially private generator. [5] suggested that it is not necessary to add noisy gradients to all weights, but only needs to achieve DP in the backpropagation process from discriminator to generator. [26] applies random projection to reduce the dimensionality of the gradient to achieve DP with less noise. [37] combined PATE with sign SGD to train generators. [3] applied differentially private optimal transmission theory to train generators. [6] proposed DPGEN which trained an energy-guided network to fit the private data. In this work, we sidestep the impact of clipping and noise through a differentially private random projection approach.

Generative model learning. With the development of generative techniques, recent works began to train generative models to generate data for downstream tasks. [13] proposed GANs, which consist of a generator and a discriminator. These two form an adversarial relationship to drive the generator to generate realistic images. Subsequently, many works [8; 2; 29] were improved based on it with the aim of increasing stability of the training process and visual quality of generated images. However, they are still less efficient and unstable in training when applied to high-dimensional image generation tasks. [35] introduced a new probabilistic generative model to fit the data distribution and then perform anneal Langevin sampling approach to obtain generated images. [16] proposed DDPM which could generate high-quality images. [34] proposed to describe the diffusion process of two distributions with stochastic differential equation. [30] combined diffusion model and autoencoders to make the generation process more stable. Inspired by these approaches, we can release generated data instead of private data to protect privacy.

3 Backgrounds

Energy-based model (EBM) captures dependencies by associating a probability density function to each configuration of the given variables. Given a known data distribution $T(x)$, we want to fit it with a probabilistic model $E(\theta; x) = \exp(-H(\theta; x))/Z_\theta$, where $H(\theta; x)$ is an energy function with parameter θ, and $Z_\theta = \int \exp(-H(\theta; x))dx$ is a normalization constant because $E(\theta; \cdot)$ denotes a probability density function, which is also called energy distribution. Because the normalization constant Z_θ is difficult to calculate explicitly, we often use maximum likelihood estimation that avoids computing Z_θ to estimate the parameter θ from $T(x)$. We aim to maximize the log likelihood $\mathbb{E}_{x\sim T(x)}[\log E(\theta; x)]$, which is equivalent to minimizing $\mathbb{E}_{x\sim T(x)}[-\log E(\theta; x)]$. Similarly, EBM can be extended to the multivariate case.

Differential privacy (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 $\mathcal{O} \subseteq \mathcal{A}$ and any

adjacent datasets D and D' :

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

where adjacent datasets D and D' differ from each other with only one training example. ε is privacy budget, where the smaller is better, and δ is the failure probability of the mechanism $\mathcal{R}$. In our case, RR enables the EBM to satisfy $(\varepsilon, 0)$-DP (or ε-DP). Notably, DP is featured by post-processing theorem and paralleEBMl 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. We note that if each mechanism $\mathcal{R}_i$ satisfies different $(\varepsilon_i, \delta_i)$-DP, the sequence of outputs $\{\mathcal{R}_i(D_i)\}_{i=1}^n$ will satisfy $(\max_i \varepsilon_i, \max_i \delta_i)$-DP.

4 Our Approach

In this section, we present our approach DPPM in two aspects. Different from most other approaches that train generators to generate images directly, we fit the distribution of the private data $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$ with an EBM and then sampling with Hamiltonian dynamics to generate images.

4.1 Learning Differentially Private Probabilistic Models

We aim to fit the probability density function of the given dataset with an EBM, which is also equivalent to matching the gradient of energy function of dataset mentioned before. The total energy of the given system $T(\mathbf{u}, \mathbf{p}) = \exp(-(P(\mathbf{u}) + K(\mathbf{p})))/Z$ is divided into potential energy $P(\mathbf{u})$ and kinetic energy $K(\mathbf{p})$ as described in Hamiltonian dynamics, in which Z is a normalization constant. Based on this description, we use an EBM $E(\theta; \mathbf{u}, \mathbf{p}) = \exp(-H(\theta; \mathbf{u}, \mathbf{p}))/Z_\theta = \exp(-(Q(\theta; \mathbf{u}) + G(\theta; \mathbf{p})))/Z_\theta$ to estimate the energy $T(\mathbf{u}, \mathbf{p})$ of the private data system, where $Q(\theta; \mathbf{u})$ and $G(\theta; \mathbf{p})$ represent the potential energy and kinetic energy of the system, respectively. In our case, in order to make full use of the given dataset (use of label information) while enabling the generated data to carry labels for downstream tasks, we input the label data y_i through an embedding layer with fixed parameters and then concatenate it with the image data $\mathbf{x}_i$ to obtain $\mathbf{u}_i$ as shown in Fig. 1. In subsequent Hamiltonian dynamics, only the gradient of the logarithmic likelihood of energy $T(\mathbf{u}, \mathbf{p})$ is required, and also to avoid calculating Z_θ, we fit $\nabla \log T(\mathbf{u}, \mathbf{p})$ with $\nabla \log E(\theta; \mathbf{u}, \mathbf{p})$ directly as mentioned in section 3. $\nabla \log E(\theta; \mathbf{u}, \mathbf{p})$ can be described as follows,

$$\nabla \log E(\theta; \mathbf{u}, \mathbf{p}) = -\nabla Q(\theta; \mathbf{u}) - \nabla G(\theta; \mathbf{p}) + \nabla \log Z_\theta, \tag{2}$$

where Z_θ is a constant so $\nabla \log Z_\theta = 0$. To simplify the calculation, we assume that the kinetic energy $K(\mathbf{p})$ is known and $\mathbf{p}$ is sampled in a known distribution such as Gaussian distribution. From a physical point of view, kinetic energy is proportional to the square of the velocity $\mathbf{p}^\top \mathbf{p}/2$, so the kinetic energy is considered to be unaffected by parameter θ and briefly write it as $G(\mathbf{p})$. We emphasize that $G(\mathbf{p}) = K(\mathbf{p})$ and it is not necessary to know the exact form of $G(\mathbf{p})$, only that $G(\mathbf{p}) \propto \mathbf{p}^\top \mathbf{p}/2$. The coefficients could be considered to be merged into the sampling step size. Therefore, only the potential energy in the EBM system needs to be fitted and the gradient of its training loss function can be expressed as follows according to [21]:

$$\nabla \mathcal{L}(\theta; \mathbf{u}) = \mathbb{E}_{\mathbf{u} \sim P(\mathbf{u})} \left[\nabla Q(\theta; \mathbf{u})\right] - \mathbb{E}_{\mathbf{u} \sim Q(\theta; \mathbf{u})} \left[\nabla Q(\theta; \mathbf{u})\right]. \tag{3}$$

Then, the parameters θ can be updated by gradient descent $\theta \leftarrow \theta + \gamma \cdot \nabla \mathcal{L}(\theta; \mathbf{u})$.

Specifically, since only the logarithmic gradient of the data distribution which is called $score$ is used in the sampling process to generate images, we model $\nabla P(\mathbf{u})$ with a NN ($\nabla Q(\theta; \mathbf{u})$ is the output of NN). For the loss function, we use the Fisher divergence to optimize the network and it can be formulated as follows:

$$\mathcal{D}_F\left(P(\mathbf{u}) \| Q(\theta; \mathbf{u})\right) = \mathbb{E}_{\mathbf{u} \sim P(\mathbf{u})} \left[\frac{1}{2} \left\|\nabla P(\mathbf{u}) - \nabla Q(\theta; \mathbf{u})\right\|^2\right]. \tag{4}$$

It can be found that in the backpropagation process, we need to compute the Hessian matrices of $P(\mathbf{u})$ and $Q(\theta; \mathbf{u})$. Due to the high dimensionality, it relies heavily on arithmetic power in the calculation, so we use the random projection method to reduce its dimensionality according to [33]. First we

sample a projection vector $\mathbf{v}$ from a known distribution, such as a Gaussian distribution. Then project the gradients of $P(\mathbf{u})$ and $Q(\theta;\mathbf{u})$ into the direction of projection vector $\mathbf{v}$. Finally, compute the loss function as follows:

$$\mathcal{L}\left(\theta;\mathbf{u},\mathbf{v}\right)=\mathbb{E}_{\mathbf{u}\sim P(\mathbf{u})}\left[\frac{1}{2}\left\|\mathbf{v}^{\top}\nabla P(\mathbf{u})-\mathbf{v}^{\top}\nabla Q(\theta;\mathbf{u})\right\|^{2}\right]. \tag{5}$$

Training the neural network with the logarithmic gradient of private data $\nabla P(\mathbf{u})$ directly may pose a privacy leakage risk. RR is performed to ensure the network does not leak sensitive information. It perturbs the training data so that trained network does not reveal the true position of private data during the sampling process. Therefore, we can release the trained network without concern about compromising privacy, because it is difficult for adversaries to tell whether an image is in the training data or not. In this work, we aim to privatize the gradient of potential energy function $\nabla P(\mathbf{u})$ in Eq. (4), which is also equivalent to privatize $\mathbf{v}^{\top}\nabla P(\mathbf{u})$ in Eq. (5). Specifically, for a batch of processed data $\mathbf{u}=\{\mathbf{u}_i\}_{i=1}^{b}$ and its projection vector $\mathbf{v}=\{\mathbf{v}_i\}_{i=1}^{b}$, they correspond to each other, so the projection vectors $\mathbf{v}_i$ of $\nabla Q(\theta;\mathbf{u}_i)$ and $\nabla P(\mathbf{u}_i)$ are the same without any protection. In our case, we apply a RR mechanism $\mathcal{R}(\cdot)$ to perturb the projection vector of $\nabla P(\mathbf{u}_i)$, which protects the private information by making the projection vectors of $\nabla P(\mathbf{u})$ and $\nabla Q(\theta;\mathbf{u})$ not necessarily strictly identical. This perturbation privatizes $\mathbf{v}^{\top}\nabla P(\mathbf{u})$, which indirectly randomizes true directions toward perceptually realistic images. The RR mechanism $\mathcal{R}(\cdot)$ can be formulated as follows:

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

where $\mathbf{v}^{-}$ is a subset of $\mathbf{v}$, $k=|\mathbf{v}^{-}|$ and $k\geq 2$. k is a hyperparameter, and we choose the first k projection vectors with the smallest cosine distance from $\mathbf{v}_i$ to $\mathbf{v}$. We note that $\mathcal{R}(\cdot)$ achieves pure DP, which does not have a failure probability compared to most of other differentially private generative approaches. After privatizing $\nabla P(\mathbf{u})$, another problem is the difficulty in obtaining $P(\mathbf{u})$. Usually, we do not know the exact form of the given data distribution, but we assume that as training proceeds, $\nabla Q(\theta;\mathbf{u})$ converges to $\nabla P(\mathbf{u})$ and both satisfy some weak regularization conditions as mentioned in [17]. So combined with [33], our loss function can be formulated as follows:

$$\begin{aligned}\mathcal{L}\left(\theta;\mathbf{u},\mathbf{v},\mathbf{v}^{r}\right)&=\mathbb{E}_{\mathbf{u}\sim P(\mathbf{u})}\left[\frac{1}{2}\left\|\mathbf{v}^{r\top}\nabla P(\mathbf{u})-\mathbf{v}^{\top}\nabla Q(\theta;\mathbf{u})\right\|^{2}\right]\\&=\mathbb{E}_{\mathbf{u}\sim P(\mathbf{u})}\left[\mathbf{v}^{r\top}\nabla^{2}Q(\theta;\mathbf{u})\mathbf{v}^{r}+\frac{1}{2}\left(\mathbf{v}^{\top}\nabla Q(\theta;\mathbf{u})\right)^{2}\right]+\text{const},\end{aligned} \tag{7}$$

where $\mathbf{v}^{r}=\mathcal{R}(\mathbf{v})$. The complete learning process of differentially private probabilistic models is as follows, we first sample a batch of data $\{\mathbf{x}_i,y_i\}_{i=1}^{b}$ and embed the labels to the images to obtain $\{\mathbf{u}_i\}_{i=1}^{b}$. Then the random vectors $\{\mathbf{v}_i\}_{i=1}^{b}$ are sampled from a Gaussian distribution and for each vector $\mathbf{v}_i$ the top-k nearest vectors are selected to compute the perturbed projection vectors $\{\mathbf{v}_i^{r}\}_{i=1}^{b}$ with RR mechanism. Finally, we train the network with $\{\mathbf{u},\mathbf{v},\mathbf{v}^{r}\}$ and loss function described in Eq. (7). Although we only privatize post-projection vector, we can think of the training process and sampling process as post-processing functions, such that the trained network and the generated data also satisfies DP according to the post-processing theorem of DP. Overall, we can obtain the following Theorem. 1, which is proved in appendix.

Theorem 1 *Our DPPM satisfies ε-DP.*

4.2 Sampling with Hamiltonian Dynamics

We perform MCMC sampling with leap frog method of Hamiltonian dynamics to generate images after the training process of NN is completed. We assume that the data distribution is $E(\theta;\mathbf{u},\mathbf{p})=$

$\exp(-(Q(\theta;\mathbf{u})+G(\theta;\mathbf{p})))/Z_\theta$, so the sampling process could be described as follows:

$$\begin{aligned}\mathbf{p}\left(t+\frac{\lambda}{2}\right)&=\mathbf{p}(t)-\frac{\lambda}{2}\nabla_{\mathbf{u}}\log E(\theta;\mathbf{u}(t),\mathbf{p}(t))\\ \mathbf{u}(t+\lambda)&=\mathbf{u}(t)+\lambda\nabla_{\mathbf{p}}\log E(\theta;\mathbf{u}(t),\mathbf{p}\left(t+\frac{\lambda}{2}\right)),\\ \mathbf{p}(t+\lambda)&=\mathbf{p}\left(t+\frac{\lambda}{2}\right)-\frac{\lambda}{2}\nabla_{\mathbf{u}}\log E(\theta;\mathbf{u}(t+\lambda),\mathbf{p}(t+\frac{\lambda}{2}))\end{aligned} \tag{8}$$

where λ is step size. $\nabla_{\mathbf{u}}\log E(\theta;\mathbf{u},\mathbf{p})=\nabla_{\mathbf{u}}Q(\theta;\mathbf{u})$ which is the output of the trained NN and $\nabla_{\mathbf{p}}\log E(\theta;\mathbf{u},\mathbf{p})=\nabla_{\mathbf{p}}G(\mathbf{p})$. As we mentioned earlier, the kinetic energy of the system is proportional to the square of the velocity $G(\mathbf{p})\propto\mathbf{p}^\top\mathbf{p}/2$, so $\nabla_{\mathbf{p}}G(\mathbf{p})=\alpha\cdot\mathbf{p}$. α is the scale factor, which could be combined into the step size λ, so we consider $\nabla_{\mathbf{p}}G(\mathbf{p})=\mathbf{p}$. In this way, the distribution of $\mathbf{u}(T)$ will converge infinitely to $P(\mathbf{u})$ when step size λ converge to 0 and $T\to\infty$, in which case $\mathbf{u}(T)$ could be considered an exact sample from $P(\mathbf{u})$ under some regularity conditions [39]. In addition to performing the above sampling process, after a certain number of iterations, we use Metropolis Guidelines to decide whether to accept the latest state of the image. The acceptance probability is $\min\left(1,E(\theta;\mathbf{u}(t+1),\mathbf{p}(t+1))/E(\theta;\mathbf{u}(t),\mathbf{p}(t))\right)$. This allows the sampling process to explore every state of distribution of private data $P(\mathbf{u})$ to produce more realistic images. We note that in some complex practical scenarios, especially when T is large, we can ignore this process [7; 28]. Inspired by the setting of learning rate in machine learning, to speed up the sampling process, we adopt strategy step size decay. Initially, a larger step size is used to make it move quickly near the realistic images, and then a smaller step size is used to fill in the image details. After the data sampling process is completed, we obtain the generated images and labels $\{\mathbf{x}'_i,y'_i\}_{i=1}^b$ for downstream tasks by the inverse of the embedding process $\{\mathbf{u}'_i\}_{i=1}^b\to\{\mathbf{x}'_i,y'_i\}_{i=1}^b$.

5 Experiments

To verify the effectiveness of our proposed differentially private probabilistic models learning approach **DPPM**, we compare it with 9 state-of-the-art differentially private generative approaches and evaluate the data utility and visual quality on four image datasets, which contain two high-resolution datasets CelebA [25] and LSUN[43]. To make the comparisons fair, our experiments use the same experimental settings as these baselines and take results from their original papers.

5.1 Experimental Setup

In this section, we briefly describe some experimental settings. More experimental details such as model architecture can be found in supplementary material.

Datasets. We conduct experiments on four benchmarking datasets, including MNIST [22], Fashion-MNIST [40], CelebA and LSUN. CelebA-H and CelebA-G are created based on CelebA with hair color (black/blonde/brown) and gender as the label, respectively. For LSUN, we choose bedroom category to conduct experiments.

Baselines. We compare our DPPM with 9 state-of-the-art approaches, including DP-GAN [41], PATE-GAN [18], DP-MERF [14], GS-WGAN [5], P3GM [36], G-PATE [26], DataLens [37], DPGEN [6] and PSG [4].

Metrics. We evaluate our DPPM as well as baselines in terms of classification performance and perceptual scores under the same different privacy budget constraints. In particular, the classification performance 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 generated images.

5.2 Experimental Results

Classification performance comparisons. To demonstrate the high data utility of our approach, we conduct comparisons with 9 state-of-the-art baselines under two privacy budget setting $\varepsilon=1$ and

Table 1: Accuracy comparisons with 9 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
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
PSG (NeurIPS'22)	0.8090	**0.9560**	0.7020	0.7770	N/A	N/A	N/A	N/A
DPPM (Ours)	**0.9215**	0.9513	**0.8309**	**0.8893**	**0.7555**	**0.8918**	**0.6954**	**0.8345**

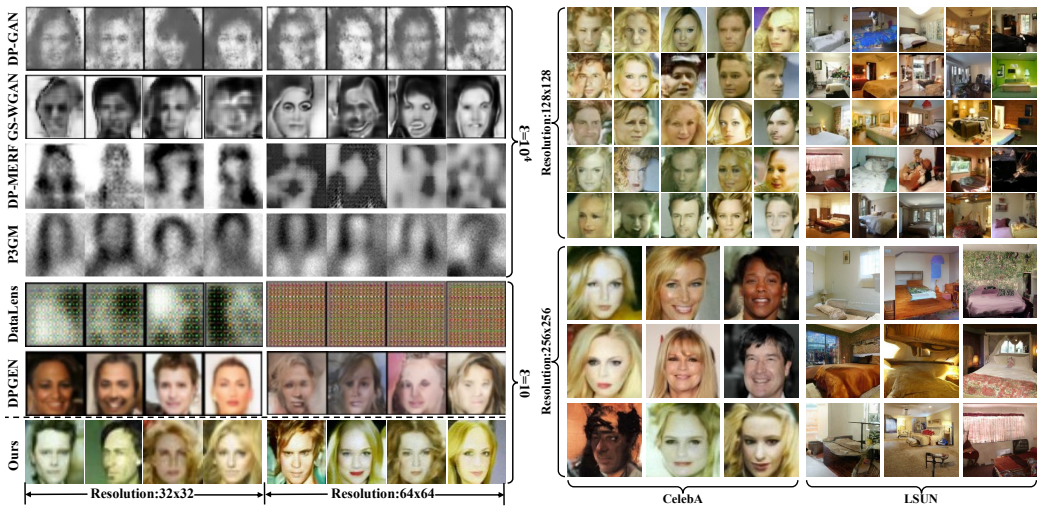
Table 2: Perceptual scores comparisons with 8 state-of-the-art baselines on CelebA 64×64 under different privacy budget ε.

Approach	ε	**IS**$\uparrow$	**FID**$\downarrow$
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
DPPM (Ours)	10	**1.53**	**52.452**

$\varepsilon = 10$ on MNIST, FashionMNIST, CelebA-H and CelebA-G. All other baselines except DPGEN are under a low failure probability $\delta = 10^{-5}$ and both DPGEN and our DPPM are under failure probability $\delta = 0$. The results are shown in Tab. 1. The performance is evaluated with test accuracy of a classifier trained with the generated data. We can find our approach achieves the highest test accuracy under the same condition of privacy budget in most cases. In particular, when ε=1, the performance of most approaches is below 80%, while our DPPM can reach 92% on MNIST and 83% on FashionMNIST. Even for higher resolution image datasets CelebA-H and CelebA-G, our approach still achieves the highest accuracy, which demonstrates its advantages in higher dimensional data over other differentially private generative approaches. These results imply that our approach could generate efficient and useful high-resolution images.

Perceptual comparisons. In addition, to demonstrate the high quality of the data generated with our approach, we evaluate it with two metrics: IS and FID, as mentioned before. Since there is no corresponding experimental results for PSG, we compared the remaining 8 approaches. The results are shown in Tab. 2. The larger IS means better quality and FID is the opposite of IS. Compare with other baselines, our approach achieves the highest IS of 1.53 and the lowest FID of 52.452 under the lowest privacy budget of 10. The main reason comes from two aspects. On one hand, we achieve DP by applying RR instead of clipping and adding noise to the gradients, avoiding direct damage to the gradients. On the other, EBM has better stability during the training process compared to GANs. In addition, these results show the robustness of our approach, even with the randomized response perturbation, the trained network could still predict the position of the realistic images correctly.

Visual comparisons of generated data. We visualize the generated data to further demonstrate the scalability and effectiveness of our approach. We compare the visualization results with other



(a) (b)

Figure 2: (a) Visualization results of DP-GAN, GS-WGAN, DP-MERF, P3GM, DataLens, DPGEN and our DPPM on CelebA with 32×32 and 64×64 resolutions. (b) Visualization results of images with 128×128 and 256×256 resolutions.

Table 3: Perceptual scores (IS↑/FID↓) on high resolution datasets under different privacy budget ε.

Dataset	Resolution	$\varepsilon = 5$	$\varepsilon = 10$	$\varepsilon = 20$
CelebA	128×128	1.25/80.259	1.30/56.160	1.50/53.398
	256×256	1.17/83.397	1.21/57.291	1.44/54.976
LSUN	128×128	2.52/154.88	2.57/88.527	2.63/79.130
	256×256	2.49/220.13	2.51/142.46	2.55/99.258

baselines and the results are shown in Fig. 2(a). We can find that even under such a luxury privacy budget condition as $\varepsilon = 10^4$, the grayscale images generated by DP-GAN, GS-WGAN, DP-MERF and P3GM are still blurry. We note that grayscale images have lower dimensionality than color images, which means it is easier to balance data quality with privacy protection. The color images generated by DPGEN have better visual quality than DataLens, but facial details are not fleshed out. In contrast, the images generated by DPPM are more realistic and have more complete facial details.

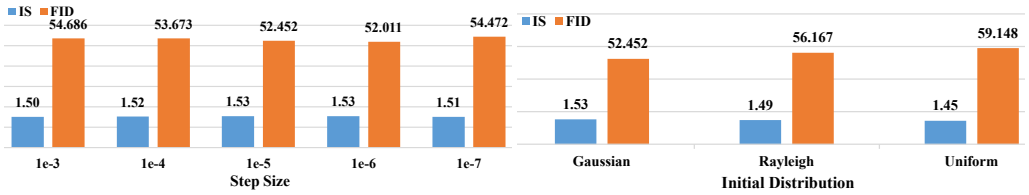
Higher-resolution images generation. To demonstrate that our approach can generate higher-resolution images well, we conduct experiments on two datasets CelebA and LSUN under different privacy budget. The results are shown in Tab. 3. As we expected, the quality of the generated images gets better as we increase ε. This is because as ε increases, the effect of perturbation caused by randomized response is decreasing, and the distribution of the generated images becomes closer to the original data. We can also find that as the resolution increased, the perceptual scores of the generated images decreased. We speculate that this is because higher-resolution images require more information about the private data distribution. In order to show the effectiveness of our approach more intuitively, we also visualize the images on CelebA and LSUN in both 128×128 and 256×256 resolutions and the results are shown in Fig. 2(b). From the visualization results we find that although the images with 256×256 resolution have lower scores, they seem to have better visual quality and contain more details such as background. We also notice that the frequency of error results (the image on the bottom left) is higher for the images with higher resolution during the generation process.

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 distribution of kinetic energy.

Table 4: Perceptual scores (IS↑/FID↓) on datasets under different resolution and k ($\varepsilon = 10$).

Dataset	Resolution	$k = 5$	$k = 10$	$k = 20$
CelebA	128×128	1.29/55.341	1.30/56.160	1.33/80.129
	256×256	1.18/56.981	1.21/57.291	1.23/97.126
LSUN	128×128	2.54/80.113	2.57/88.527	2.58/117.24
	256×256	2.49/121.72	2.51/142.46	2.53/154.64



(a) (b)

Figure 3: Perceptual scores on CelebA at 64×64 resolution under different step size (a) and kinetic energy distribution (b).

Impact of k. To study how the hyperparameter k affects the trade-off between privacy and data utility, we compare the perceptual scores when k takes different values under the same privacy budget $\varepsilon = 10$. Eq. (6) shows that the smaller k is, the higher the probability of $\mathcal{R}(\mathbf{v}_i) = \mathbf{v}_i$ is, which will cause the corresponding generated images are closer to the distribution of original private data. FID is consistent with this rule, but we find that IS increases with increasing k. We suspect that this is related to the IS itself. IS does not reflect the distance between the generated data and the original private data; it measures the entropy of the distribution of the generated data, focusing on its diversity. The increase of k will enlarge the diversity of the generated data (more data appearing that do not match the distribution of original private data), which leads to an increase in the IS score.

Impact of step size. To study the step size λ effect on the visual quality and the data utility of the generated images, we generate images with different step size λ and compare their perceptual scores. The results are shown in Fig. 3(a). We find that decreasing the step size λ weakly improves the quality of the generated images when λ is less than $1e-6$. When $\lambda = 1e-7$, there is a slight decrease in quality of the generated images compared to $\lambda = 1e-6$. Similar to the learning rate parameter in machine learning, the smaller step size λ is, the smaller the moving distance per sample is, which can fill more details to the images, but if λ is too small, the image will fall into local optimum. This is the reason why the results in Fig. 3(a) appear.

Impact of kinetic energy distribution. The initial distribution of kinetic energy $G(\mathbf{p})$ which is also the initial distribution of $\mathbf{p}$ is an important factor of Hamiltonian dynamics. We sample images with different initial distribution (Gaussian, Rayleigh and Uniform distributions) of $\mathbf{p}$ and show the results in Fig. 3(b). We can find that the images generated with the initial value of $\mathbf{p}$ sampled from the Gaussian distribution have the best quality. Rayleigh distribution is the joint distribution of two independent Gaussian distributions, and the quality of the images generated by sampling from it is slightly inferior to that of the Gaussian distribution. The image quality is worst when the initial value of $\mathbf{p}$ is sampled from Uniform distribution, but the difference in quality between the images obtained when the initial value of $\mathbf{p}$ was sampled from the three distributions is not very large. This is because from the energy point of view, kinetic and potential energy are conserved when a system is not subject to external action. We can also know from Eq. (8) that the two are interacting with each other, and the different distributions only affect the magnitude of the initial energy of the system.

6 Conclusion

The releasing of private data or networks trained on it may pose a privacy leakage risk. To allow for secure deployment, we propose DPPM to generate differentially private images for releasing. DPPM

fits the logarithmic gradient of private data distribution, which serves as a movement direction toward realistic images in Hamiltonian dynamics. During the training process, we perturb the projection vectors with RR mechanism to achieve DP protection. Thus, this trained network can be released to provide the movement directions, which are used in the process of image generation with Hamiltonian MCMC sampling, in a differentially private manner. We note that DPPM satisfies pure DP, which does not have a failure probability compared to most other differentially private generative approaches. Extensive experiments and privacy analysis are conducted to show the effectiveness of our approach.

References

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. B. McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[3] Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence. In *Advances in Neural Information Processing Systems*, volume 34, pages 12480–12492, 2021.

[4] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. Private set generation with discriminative information. In *Advances in Neural Information Processing Systems*, 2022.

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

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

[7] Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, 2014.

[8] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2180–2188, 2016.

[9] Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *ArXiv:2210.09929*, 2022.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284, 2006.

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

[12] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[14] Frederik Harder, Kamil Adamczewski, and Mijung Park. Dp-merf: Differentially private mean embeddings with random features for practical privacy-preserving data generation. In *International Conference on Artificial Intelligence and Statistics*, pages 1819–1827, 2021.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020.

[17] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[18] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.

[19] Anthony D. Kennedy. Hybrid monte carlo. In *Physics Letters B*, pages 216–222, 1987.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.

[21] Rithesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *ArXiv:1901.08508*, 2019.

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.

[23] Yann LeCun, Sumit Chopra, Raia Hadsell, Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press, 2006.

[24] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5168–5177, 2017.

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

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

[27] Radford M. Neal. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, pages 112–172, 2011.

[28] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *Association for the Advancement of Artificial Intelligence*, 2020.

[29] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*, pages 2642–2651, 2017.

[30] Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. VAEs meet diffusion models: efficient and high-fidelity generation. In *Advances in Neural Information Processing Systems Workshop*, 2021.

[31] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[33] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Conference on Uncertainty in Artificial Intelligence*, 2019.

[34] Yang Song, Jascha Soh-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.

[35] Yang Song and Ermon Stefano. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

[36] Shun Takagi, Tsubasa Takahashi, Yang Cao, and Masatoshi Yoshikawa. P3gm: Private high-dimensional data release via privacy preserving phased generative model. In *International Conference On Data Engineering*, pages 169–180, 2021.

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

[38] Stanley L. Warner. Randomized response: a survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[39] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning*, 2011.

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

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

[42] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 225–240, 2019.

[43] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *ArXiv:1506.03365*, 2015.

Appendix

Procedure of DPPM

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

Alg. 1 is the training procedure of differentially private probabilistic models. It can be simply described as the following five steps:

- randomly sample a batch of data : $\{\mathbf{x}_i, y_i\}_{i=1}^b$.
- compute $\mathbf{u} = \{\mathbf{u}_i\}_{i=1}^b$ with $\{\mathbf{x}_i, y_i\}_{i=1}^b$
- randomly sample a batch of vectors $\mathbf{v} = \{\mathbf{v}_i\}_{i=1}^b$.
- form $\mathbf{v}^-$ and compute $\mathbf{v}^r = \mathcal{R}(\mathbf{v}|\mathbf{v}^-)$.
- calculate the loss function with Eq. (7) and update the network.

We note that because the network is used to predict the movement direction of the sampling, we choose the cosine distance to select the top-k nearest projection vectors to form $\mathbf{v}^-$.

Alg. 2 is the sampling procedure of generated images. Different from traditional Hamiltonian dynamics, benefits from the setting of learning rate in machine learning, 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 Differentially Private Probabilistic Model Learning

Input: Private data D, number of training iterations T, loss function $\mathcal{L}(\theta; \cdot)$, randomized response 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 $\{\mathbf{x}_i, y_i\}_{i=1}^b$ from D
4: Compute $\mathbf{u} = \{\mathbf{u}_i\}_{i=1}^b$ with $\{\mathbf{x}_i, y_i\}_{i=1}^b$
5: Sample a batch of vectors $\mathbf{v} = \{\mathbf{v}_i\}_{i=1}^b$ from a Gaussian distribution
6: $\mathbf{v}^r = \Phi$
7: **for** each $\mathbf{v}_i$ in $\mathbf{v}$ **do**
8: Select top-k nearest vectors to $\mathbf{v}_i$ from $\mathbf{v}$ form $\mathbf{v}^-$
9: Calculate $\mathbf{v}_o = \mathcal{R}(\mathbf{v}_i)$ given $\mathbf{v}^-$
10: $\mathbf{v}^r = \mathbf{v}^r \cup \{\mathbf{v}_o\}$
11: **end for**
12: Calculate loss $\mathcal{L}(\theta; \mathbf{u}, \mathbf{v}, \mathbf{v}^r)$
13: Update the network $\theta_{t+1} \leftarrow \theta_t + \gamma \nabla \mathcal{L}$
14: **end for**
15: **return** θ_T

Proof of Theorem. 1

Recall the loss function introduced in the Eq. (7), we need the triplet dataset $\{\mathbf{u}, \mathbf{v}, \mathbf{v}^r\}$ as input. Given a batch of data $D = \{\mathbf{x}_i, y_i\}_{i=1}^b$ and projection vector $\mathbf{v} = \{\mathbf{v}_i\}_{i=1}^b$, the probability of $\mathcal{L}\left(\theta; \mathbf{u}_i, \mathbf{v}_i, \mathbf{v}_i^r\right)$ is as follows:

$$Pr[\mathbf{u}_i, \mathbf{v}_i, \mathbf{v}_i^r, \mathbf{v}^-|\mathbf{v}, D] = Pr[\mathbf{u}_i|D] \cdot Pr[\mathbf{v}_i|\mathbf{v}] \cdot Pr[\mathbf{v}^-|\mathbf{v}_i, \mathbf{v}] \cdot Pr[\mathcal{R}(\mathbf{v}_i) = \mathbf{v}_i^r|\mathbf{v}^-]. \quad (8)$$

This equation is obtained by Bayes' theorem. In our approach, we sample data $\mathbf{x}_i$ and $\mathbf{v}_i$ uniformly, so that $Pr[\mathbf{u}_i|D]$ and $Pr[\mathbf{v}_i|\mathbf{v}]$ are $1/b$. Given $\mathbf{v}$ and $\mathbf{v}_i$, $\mathbf{v}^-$ is fixed, so $Pr[\mathbf{v}^-|\mathbf{v}_i, \mathbf{v}] = 1$. So $Pr[\mathbf{u}_i, \mathbf{v}_i, \mathbf{v}_i^r, \mathbf{v}^-|\mathbf{v}, D] = Pr[\mathcal{R}(\mathbf{v}_i) = \mathbf{v}_i^r|\mathbf{v}^-] \cdot 1/b^2$.

Given a image $\mathbf{x}_i$ and its projection vector $\mathbf{v}_i$, we define $\mathcal{M}(\mathbf{v}_i, \mathbf{u}_i) = \mathcal{R}(\mathbf{v}_i) \cdot \mathcal{H}(\mathbf{u}_i) = \mathbf{v}_i^\top \cdot \nabla Q(\theta; \mathbf{u}_i)$, we can find that $\mathcal{R}$ and $\mathcal{H}$ are independent of each other, so $Pr[\mathcal{M}(\cdot)] = Pr[\mathcal{R}(\cdot)] \cdot$

Algorithm 2 Sampling with Hamiltonian Dynamics

Input: Trained network $\nabla Q(\theta;\cdot)$, kinetic energy $G(p)$, step size λ_0, private data D, sampling iterations M, acceptance-rejection iterations N

Initialize $\mathbf{x}(0)$ randomly
for $m \in [M]$ **do**
  Initialize $p(m)$ randomly
  $\lambda = \lambda_0 \cdot (M/m)^2$
  for $n \in [N]$ **do**
    $p\left(m+\frac{\lambda}{2}\right) = p(t) - \frac{\lambda}{2}\nabla Q(\theta;\mathbf{x}(m))$
    $\mathbf{x}(m+\lambda) = \mathbf{x}(m) + \lambda \nabla G\left(p\left(m+\frac{\lambda}{2}\right)\right)$
    $p(m+\lambda) = p\left(m+\frac{\lambda}{2}\right) - \frac{\lambda}{2}\nabla Q(\theta;\mathbf{x}(m+\lambda))$
  end for
  Calculate the probability $p_a = \min(1, E(\theta;\mathbf{x}(m+N\lambda),\mathbf{p}(m+N\lambda))/E(\theta;\mathbf{x}(m),\mathbf{p}(m)))$
  $\mathbf{x}(m+1) = \mathbf{x}(m+N\lambda)$ with probability and $\mathbf{x}(m+1) = \mathbf{x}(m)$ with $1-p_a$
end for
return $\mathbf{x}(M)$

$Pr[\mathcal{H}(\cdot)]$. In the attack scenario, we pair $(\mathbf{x}_i, y_i)$ and $\mathbf{v}_i$ actually, but attacker thinks $(\mathbf{x}_i, y_i)$ is paired with $\mathbf{v}_j$. In our approach, as soon as $(\mathbf{x}_i, y_i)$ is determined, the corresponding $\mathbf{v}_i$ and $\mathbf{v}^-$ are already determined, so $Pr[\mathbf{v}^-|(\mathbf{x}_i, y_i), \mathbf{v}_i] = Pr[\mathbf{v}^-|(\mathbf{x}_i, y_i), \mathbf{v}_j]$. For simplicity, in the following we default to $\mathbf{v}^-$ being the same and omit it.

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

$$Pr\left[\mathcal{M}\left(\mathbf{v}_i, \mathbf{x}_i\right) = \mathbf{o}_i\right] \leq e^{\varepsilon} \cdot Pr\left[\mathcal{M}\left(\mathbf{v}_j, \mathbf{x}_j\right) = \mathbf{o}_i\right], \tag{9}$$

where $\mathbf{o}_i$ is a possible output of $\mathcal{M}$.

Proof *From the definition of RR (Eq. (6)), we can know the probability that $\mathcal{R}(\cdot)$ takes as input $\mathbf{v}_i$ and returns $\mathbf{v}_i$ is the largest for $e^{\varepsilon}/(e^{\varepsilon}+k-1)$ and that takes as input $\mathbf{v}_j$ and returns $\mathbf{v}_i$ is the smallest for $1/(e^{\varepsilon}+k-1)$. We sample $(\mathbf{x}_i, y_i)$ uniformly, and $(\mathbf{x}_i, y_i)$ and $\mathbf{u}_i$ are in one-to-one correspondence, so $Pr[\mathcal{H}(\mathbf{u}_i)] = Pr[\mathcal{H}(\mathbf{u}_j)]$. Then we have*

$$\begin{aligned} Pr\left[\mathcal{M}\left(\mathbf{v}_i, \mathbf{u}_i\right) = \mathbf{o}_i\right] &= Pr\left[\mathcal{R}(\mathbf{v}_i) = \mathbf{v}_o\right] \cdot Pr[\mathcal{H}(\mathbf{u}_i)] \\ &\leq e^{\varepsilon} \cdot Pr\left[\mathcal{R}(\mathbf{v}_j) = \mathbf{v}_o\right] \cdot Pr[\mathcal{H}(\mathbf{u}_i)] \\ &= e^{\varepsilon} \cdot Pr\left[\mathcal{R}(\mathbf{v}_j) = \mathbf{v}_o\right] \cdot Pr[\mathcal{H}(\mathbf{u}_j)] \\ &= e^{\varepsilon} \cdot Pr\left[\mathcal{M}\left(\mathbf{v}_j, \mathbf{u}_j\right) = \mathbf{o}_i\right] \end{aligned} \tag{10}$$

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

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

$$\begin{aligned} Pr[\mathcal{M}(\mathbf{v}, \tilde{D}) \in O] &= Pr\left[\mathcal{M}\left(\mathbf{v} \cap \mathbf{v}', \tilde{D} \cap \tilde{D}'\right) \in O\right] \cdot Pr\left[\mathcal{M}\left(\mathbf{v}_i, \mathbf{u}_i\right) = \mathbf{o}_i\right] \\ &\leq e^{\varepsilon} \cdot Pr\left[\mathcal{M}\left(\mathbf{v} \cap \mathbf{v}', \tilde{D} \cap \tilde{D}'\right) \in O\right] \cdot Pr\left[\mathcal{M}\left(\mathbf{v}_i', \mathbf{u}_i'\right) = \mathbf{o}_i\right] \\ &= e^{\varepsilon} \cdot Pr[\mathcal{M}(\mathbf{v}', \tilde{D}') \in O], \end{aligned} \tag{11}$$

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 $\mathbf{v}$. It is necessary to calculate the joint probability to clarify the association of each vector. We again note that our purpose is to generate differentially private images. Depending on the post-processing property of differential privacy, we can achieve our goal by implementing differential privacy at any point before generating the images. Next, we prove that our DPPM satisfies differential privacy by proving that the loss function in the training process satisfies differential privacy.

Theorem 1 *Our DPPM satisfies ε-DP.*

Proof *Consider two adjacent datasets $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^b$ and $D' = \{(\mathbf{x}'_i, y'_i)\}_{i=1}^b$ and their projection vectors $\mathbf{v} = \{\mathbf{v}_i\}_{i=1}^b, \mathbf{v}' = \{\mathbf{v}'_i\}_{i=1}^b$, we define $\tilde{D} = \{\mathbf{u}_i\}_{i=1}^b = \mathcal{E}(D), \tilde{D}' = \{\mathbf{u}'_i\}_{i=1}^b = \mathcal{E}(D')$ and $\mathcal{F}(\mathbf{v}, D) = \mathcal{R}(\mathbf{v})^\top \nabla \mathcal{H}(\mathcal{E}(D))\mathcal{R}(\mathbf{v}) + \frac{1}{2}\left(\mathbf{v}^\top \mathcal{H}(\mathcal{E}(D))\right)^2$. $\mathcal{E}(\cdot)$ represents the operation of embedding the labels $\{y_i\}_{i=1}^b$ into the images $\{\mathbf{x}_i\}_{i=1}^b$ to get $\{\mathbf{u}_i\}_{i=1}^b$. Then we have*

$$
\begin{aligned}
Pr[\mathcal{F}(\mathbf{v}, \mathcal{D}) \in \mathcal{O}] &= \prod_i Pr[\mathbf{u}_i, \mathbf{v}_i, \mathbf{v}_i^r, \mathbf{v}^- | \mathbf{v}, D] \\
&= \prod_i Pr[\mathcal{R}(\mathbf{v}_i) = \mathbf{v}_i^r | \mathbf{v}^-] \cdot 1/b^2 \\
&= \prod_i Pr[\mathcal{R}(\mathbf{v}_i) \cdot \mathcal{H}(\mathbf{u}_i) = \mathbf{o}_i | \mathbf{v}^-] \cdot 1/b^2 \\
&= Pr[\mathcal{M}(\mathbf{v}, \tilde{D}) \in O] \cdot Pr[\tilde{D}|D] \cdot 1/b^2 \\
&\leq e^\varepsilon \cdot Pr[\mathcal{M}(\mathbf{v}', D') \in O] \cdot Pr[\tilde{D}|D] \cdot 1/b^2 \\
&= e^\varepsilon \cdot Pr[\mathcal{F}(\mathbf{v}', \mathcal{D}') \in \mathcal{O}],
\end{aligned}
\tag{12}
$$

where $\mathcal{O}$ 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 the loss function $\mathcal{F}(\cdot)$ of the model 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 DPPM satisfies ε-DP.

Experimental Details

Datasets. MNIST and FashionMNIST are both 10-class datasets containing 60,000 training images and 10,000 testing images. Each image is 28×28 grayscale image. CelebA is a face attribute dataset, which contains 202,599 color images of celebrity faces. We use the official preprocessed version with the face alignment and resize the images to 64×64×3 in classification performance comparison and 128×128×3, 256×256×3 in other sections. We create CelebA-H and CelebA-G based on it. CelebA-H is a classification dataset with hair color (black/blonde/brown) as the label and CelebA-G is a classification dataset with gender as the label. LSUN is a large-scale image dataset, which contains 10 scene classes and 20 object classes, totaling about 1 million color images. For LSUN, we choose the bedroom category and resize the images to 128×128×3 and 256×256×3 to evaluate the perceptual scores.

Baselines. DP-GAN is to directly apply the DPSGD training strategy to the training process of WGAN. Because WGAN itself satisfies the Lipschitz condition, the effect from gradient clipping is eliminated. PATE-GAN, DP-MERF, GS-WGAN, P3GM, G-PATE and DataLens are all based on PATE framework with different teacher aggregation strategies. All of the above baselines achieve differential privacy based on Gaussian mechanism. DPGEN achieves differential privacy based on randomized response mechanism. We get the experimental results by running official codes or from original papers. PSG incorporates the downstream task into training to improve its data quality, but it requires repeated training for different downstream tasks.

Implementations. We choose RefineNet to fit the potential energy of the system. When comparing classifier performance, we choose the same architecture as the other baselines, which is described in appendix. The default setting of k for all experiments excluding the ablation studies is 10. The training epoch of RefineNet is 10,000 for MNIST, FashionMNIST and 50,000 for CelebA, LSUN. For each dataset, we generate 10,000 samples for classifier learning. The 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.

Backbone of Models. There are two networks in our work: NN for predicting potential energy and classifier for comparison. The architecture of the NN in our framework is RefineNet (Fig. 4(a) and (b)) and the architecture of classifier model for the downstream classification task is shown in Fig. 4(c). The architecture shown in Fig. 4 (b) is used to generate 256×256 resolution images, and the architecture shown in Fig. 4 (a) is used to generate images of other resolutions (lower than 256×256).

[Conv2D, 3x3, 128]
[CondResBlock, 128] ×2
[CondResBlock, 256] ×2
[CondResBlock, 256 dilation 2] ×2
[CondResBlock, 256 dilation 4] ×2
[CondRefineBlock, 256] ×2
[CondRefineBlock, 128] ×2
[Conv2D, 3x3, 128]

(a) NN Model

[Conv2D, 3x3, 128]
[CondResBlock, 128] ×2
[CondResBlock, 256] ×2
[CondResBlock, 256 dilation 2] ×2
[CondResBlock, 256 dilation 4] ×2
[CondResBlock, 512 dilation 8] ×2
[CondRefineBlock, 512] ×2
[CondRefineBlock, 256] ×2
[CondRefineBlock, 128] ×2
[Conv2D, 3x3, 128]

(b) NN Model (For 256 × 256)

[Conv2D, 3x3, 32, ReLU]
[Max Pooling, 2x2]
[Conv2D, 3x3, 64, ReLU]
[Fully Connected, ReLU]
[Conv2D, 3x3, 64, ReLU]
[Fully Connected, ReLU]
[Fully Connected]
[Softmax Layer]

(c) Classifier Model

Figure 4: Backbone of NN model and classifier model

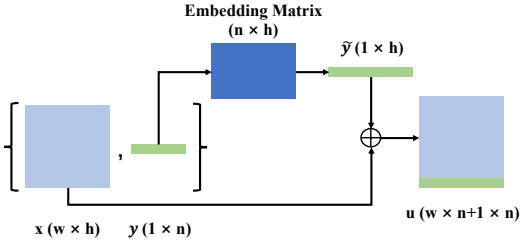


Figure 5: The procedure for embedding labels into images.

Embedding Procedure. Here, we further describe the procedure of embedding labels into images. Let's take a single data as an example and we assume that the size of the image is w×h and the size of the label is 1×n. As shown in Fig. 5, the process is divided into two steps. The first step multiplies the label with an embedding matrix to get the output of size 1×h; the second step concatenates the output obtained in the first step together with the image.

Extended Visualization Results

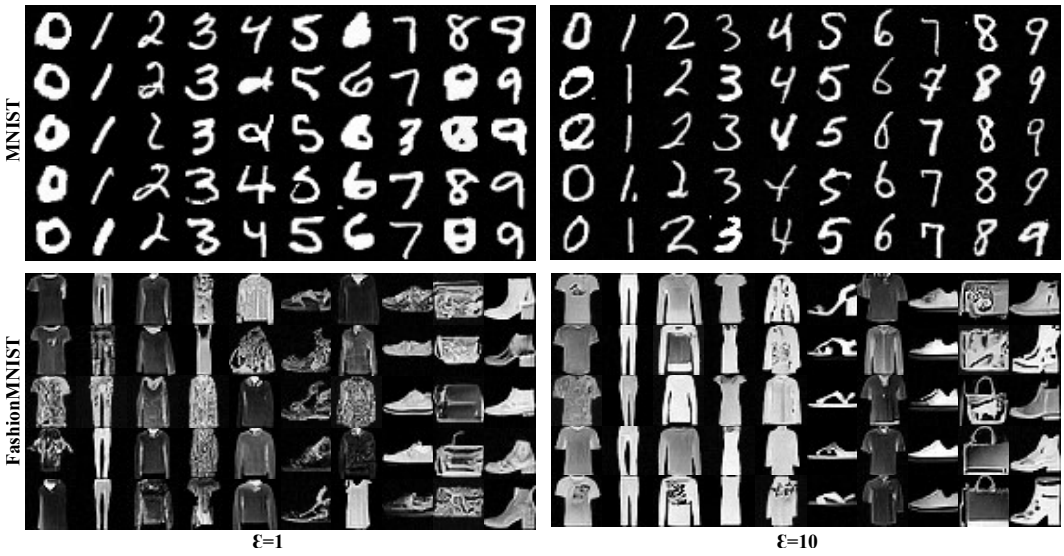


Figure 6: Visualization results of MNIST and FashionMNIST with 28×28 resolution under different privacy budget.

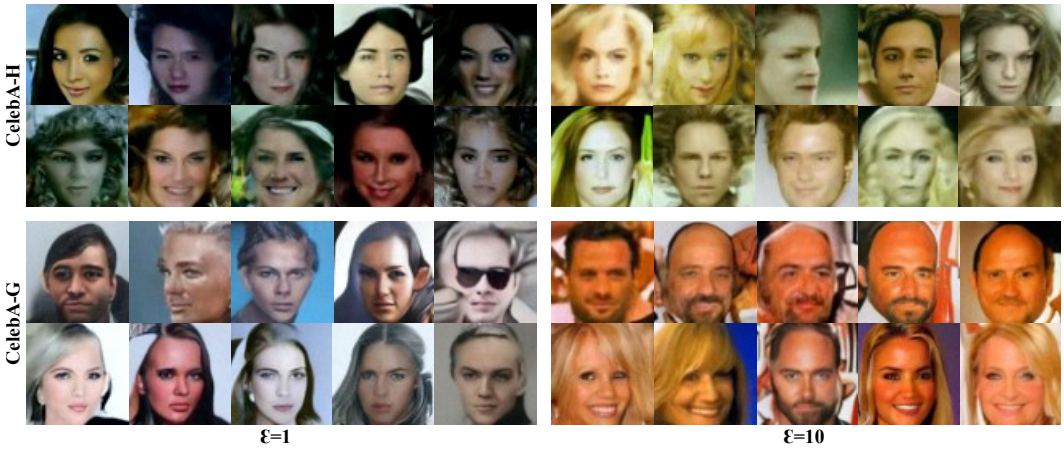


Figure 7: Visualization results of CelebA with 64×64 resolution under different privacy budget.