DPGEN: Differentially Private Generative Energy-Guided Network for Natural Image Synthesis

Jia-Wei Chen[1,2] Chia-Mu Yu[3] Ching-Chia Kao[1,2] Tzai-Wei Pang[1] Chun-Shien Lu[1,2]
[1]IIS, Academia Sinica [2]CITI, Academia Sinica [3]National Yang Ming Chiao Tung University
sk413025@gmail.com chiamuyu@gmail.com {cck123, cegg12345678}@iis.sinica.edu.tw lcs@iis.sinica.edu.tw

Abstract

Despite an increased demand for valuable data, the privacy concerns associated with sensitive datasets present a barrier to data sharing. One may use differentially private generative models to generate synthetic data. Unfortunately, generators are typically restricted to generating images of low-resolutions due to the limitation of noisy gradients. Here, we propose DPGEN, a network model designed to synthesize high-resolution natural images while satisfying differential privacy. In particular, we propose an energy-guided network trained on sanitized data to indicate the direction of the true data distribution via Langevin Markov chain Monte Carlo (MCMC) sampling method. In contrast to the state-of-the-art methods that can process only low-resolution images (e.g., MNIST and Fashion-MNIST), DPGEN can generate differentially private synthetic images with resolutions up to 128×128 *with superior visual quality and data utility. Our code is available at https://github.com/chiamuyu/DPGEN*

1. Introduction

Image synthesis (e.g., through generative adversarial networks (GANs) [16]) with differential privacy (DP) [12] can be a solution to enable data release without compromising privacy. For example, differentially private GANs (DP-GANs) [6, 47] are generally trained using differentially private stochastic gradient descent (DPSGD) [1] that perturbs the gradients in each iteration and provides an alternative to direct data release. In particular, generators from DPGANs can be made public; users can then generate synthetic data for their downstream tasks. Nevertheless, GANs have been known to be considerably difficult to train [18, 37, 44]. The situation becomes even worse when noise is introduced in DPSGD. Hence, generators are typically restricted to generating images of resolutions as low as 32×32, and are unsuitable for practical applications of image synthesis.

Difficulty in Generating High-Resolution Images from DPGAN. The gradient instability caused by interactive training from the minimax optimization of GANs is a major concern in the synthesis of high-resolution images [14, 24, 25, 26]. Nonetheless, despite various difficulties in the training of GANs, currently GANs can generate photo-realistic image of resolutions up to 1024×1024. Thus, a straightforward design of DPGANs would be applying DP to the state-of-the-art (SOTA) GANs. However, because of the increased batch size and model complexity, such a naïve combination and therefore the use of DPSGD lead to four serious problems. **1) Training Inefficiency:** Although larger batch sizes improve training stability, they also lead to a significant degradation in training efficiency (10× slower) [7] due to the per-sample gradient modification, which requires backpropagation be performed on each example in a training batch. **2) Difficulty in Tuning Hyperparameters:** Due to the complexity of neural network (NN) architectures (e.g., skip connection and attention layers), accurate estimation of global sensitivity is infeasible. This implies the occurrence of either information loss or an excessive noise scale during gradient clipping in DPSGD. **3) Large Noise Magnitude:** The increased number of dimensions required by layers in NNs to accommodate high-resolution data leads to a catastrophic amount of noise. **4) The Damage of Visual Quality from Direction of Noisy Gradient:** Despite the perceptual loss used in the backpropagation, the noisy gradient affected by the DP noise may dramatically deviate from the direction supposed to move forward, resulting in a synthesis of perceptually awful images.

In summary, when DPSGD is applied to train GANs, the techniques used by GANs for high-resolution image synthesis instead amplify the drawbacks of DPSGD.

Key Insights. The synthesis of perceptually realistic images in a DP manner is very challenging. Notably, our result is in possession of the following novelty.

From the perspective of gradient updates in the parameter space, DP noise in DPSGD completely destroys the gradients; this severely degrades the training stability. To tackle the aforementioned problems **1)**∼**4)** simultaneously, we abandon DPSGD and take a fundamentally different approach. We instead use a sampling method on the training data. More specifically, we discretize the movement directions (in terms of *MCMC* described later) and randomize the

true directions (toward perceptually realistic images) through a fixed number of carefully chosen images in a pixel space where perceptually realistic images can be easily defined to maximize training stability while preserving DP. In this way, we are guaranteed to generate perceptually realistic images and at the same time have the features preserved.

Overview of Proposed Method. Here, we propose a framework using the Markov chain Monte Carlo (MCMC) sampling method [55] to synthesize images, wherein the movement directions are guided by an energy-based network. Note that as drawing random samples takes considerable time, sampling methods such as Langevin MCMC [17, 43] or Hamiltonian Monte Carlo [11, 39] are often used to increase efficiency. To enable the above framework to satisfy DP, we propose a Differentially Private Generative Energy-guided Network (DPGEN) architecture for high-resolution image synthesis. In particular, an energy-guided network trained from DP-sanitized data helps indicate routes to sampling perceptually realistic images with high utility.

In general, DPGEN aims to privatize Langevin MCMC. The process flow of DPGEN is illustrated in Figure 1. More specifically, we privatize training images such that the sanitized data can be used to train an energy-guided network involved in the Langevin MCMC sampler. Note that all images in the sanitized dataset are visibly degraded by noise, but preserves the information about the directions of the training images. As a result, the energy-guided network trained on the sanitized dataset in a non-DP manner can synthesize perceptually realistic images by leveraging the directional information hidden in the sanitized dataset.

Contributions. The contributions are summarized below.

(a) We propose DPGEN, an instantiation of DP variant of EBM (described in Section 3), that synthesizes high-resolution images (up to 128×128 resolutions) in an ε-DP manner, in contrast to the other DPSGD-assisted GAN-based (ε, δ)-DP approaches. In fact, DPGEN can achieve the best of both worlds; i.e., it is able to synthesize perceptually realistic images that can well preserve the features such that the downstream classification task has high accuracy.

(b) Through extensive evaluations on various datasets, we demonstrate that DPGEN significantly improves the sample quality of synthetic images over SOTA approaches.

2. Related Work

Differentially Private Generative Models. [53, 54, 58] develop early-stage DPGANs for image synthesis according to DPSGD. As the quality of generated synthetic images is highly related to the noise scale in DPSGD, all of them reduce the sensitivity and thus, the noise scale, by tuning the clipping bound of the gradient norm. [23, 34] follow the PATE framework [40, 41] to derive a private generative model. [36, 46] calculate the clipping bound via adaptive clipping and moments accountants [1]. GS-WGAN [6] adopts a WGAN [2] whose 1-Lipschitz property naturally bounds the sensitivity, having a better control of noise magnitude. DP-MERF [19] synthesizes images by taking advantage of random feature representations of kernel mean embeddings. P3GM [45], a variant of private variational autoencoder with two-phased training, has more tolerance to the noise. Very recently, through the gradient compression, DataLens [50] reduces the noise scale. Based on an optimal transport-based generative model, DP-Sinkhorn [5] learns the data distribution by minimizing the Sinkhorn divergence.

Theoretical Treatments and Other Improvements. [4, 30] reduce the computation time required for each sample of gradients by replacing the auto-differentiation used in the gradient clipping from reverse-mode to forward-mode, thus allowing for a larger batch-size during the training stage. [38, 56] project the gradients to a predefined subspace so as to have a better control of the sensitivity of gradient calculation and therefore lower noise scale. [9] argues that current DPGANs overestimate the privacy loss, because the intermediate results of each training iteration are unknown to the adversary in most cases. [51, 59] assume that the elements in a dataset could be sampled uniformly, thus obtaining greater privacy through sub-sampling and privacy amplification by shuffling. [42] proposes tempered sigmoid activations as an alternative activation function to bound the sensitivity. Conversely, [8, 48] improve the utility of the models trained using DPSGD from manually designed features. [32] establishes a theoretical foundation for the employment of *warm start*, a popular technique designed to improve the utility of DPGANs. [15] applies randomized response (RR) to the case where the objective function can be nonconvex.

3. Background

Energy-based Model (EBM) [10, 29]. Given the underlying data distribution $p(x)$ of a dataset, we aim to fit $p(x)$ with a probability density model $q_\theta(x) = e^{-U_\theta(x)}/Z_\theta$, which is called the energy distribution. Here, U_θ is the energy function parameterized by θ, and $Z_\theta = \int e^{-U_\theta(x)} dx$ denotes the normalization constant (i.e., the partition function). After normalization, $q_\theta(x)$ is a probability density function (PDF).

In the estimation of parameter θ, the evaluation of Z_θ involves integration, which is difficult to calculate explicitly. Generally, the maximum likelihood estimation (MLE) is used to estimate θ from $p(x)$. The log likelihood $\mathbb{E}_{x\sim p(x)}[\log q_\theta(x)]$ is expected to be maximized, which is equivalent to minimizing a loss function $L(x;\theta) = \mathbb{E}_{x\sim p(x)}[-\log q_\theta(x)]$. From [27], we know $\nabla_\theta \log q_\theta(x) = -\nabla_\theta U_\theta(x) + \mathbb{E}_{x\sim q_\theta(x)}[\nabla_\theta U_\theta(x)]$ and derive the gradient of the loss function as

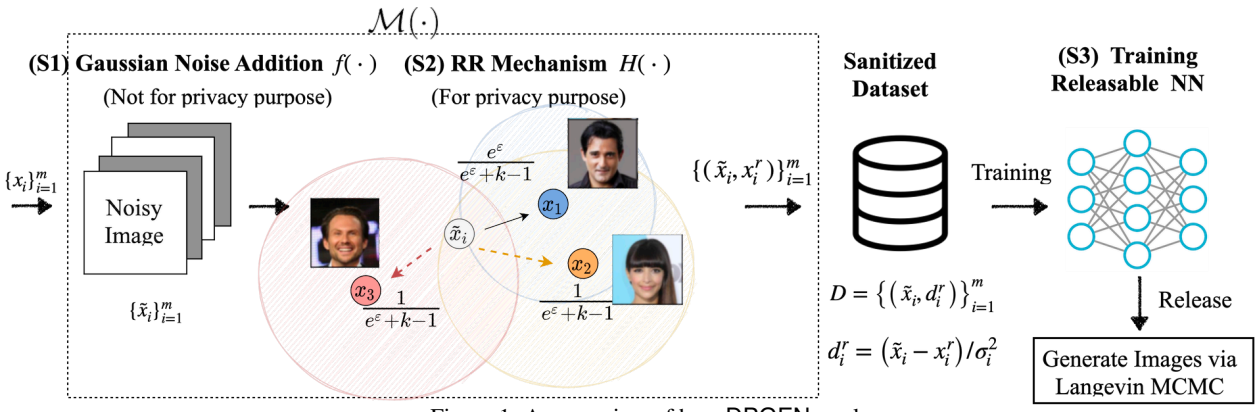


Figure 1. An overview of how DPGEN works.

$$\nabla_\theta L(x;\theta) = \mathbb{E}_{x\sim p(x)}\left[\nabla_\theta U_\theta(x)\right] - \mathbb{E}_{x\sim q_\theta(x)}\left[\nabla_\theta U_\theta(x)\right]. \quad (1)$$

Therefore, the parameters θ can be updated by gradient descent $\theta \leftarrow \theta + \gamma \cdot \nabla_\theta L(x;\theta)$.

Connecting GANs to EBM. Notably, $\nabla_\theta L(x;\theta)$ is the same as the loss function used in WGAN [2]. In evaluating $\mathbb{E}_{x\sim q_\theta(x)}\left[\nabla_\theta U_\theta(x)\right]$ in Eq. (1), it is necessary to sample x from the energy distribution $q_\theta(x)$. However, the sampling process must compute the intractable integration term Z_θ in $q_\theta(x)$. To avoid calculating Z_θ, GAN can be seen as a special case of EBM, where x is sampled by a generator $G_\varphi(z)$ in GAN with z following Gaussian distribution and with ρ as parameters, instead of being sampled from q_θ [27]. Moreover, if U_θ is a function that satisfies the 1-Lipschitz, then the parameterized function U_θ in Eq. (1) can be considered as a discriminator. Because two parameters θ and φ need to be estimated, this leads to min-max optimization and suffers from instability and difficulties in training.

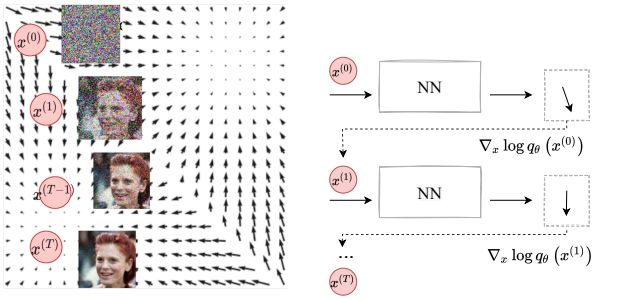
Differential Privacy (DP) [12]. A randomized mechanism $\mathcal{M}$ is (ε, δ)-DP, if

$$\Pr[\mathcal{M}(X) \in O] \leq e^\varepsilon \cdot \Pr\left[\mathcal{M}\left(X'\right) \in O\right] + \delta \quad (2)$$

holds for any adjacent datasets X and X' that differ from each other with only one training example. Here, ε is the upper bound on the privacy loss corresponding to $\mathcal{M}$, and δ is the probability of violating the DP constraint. In practice, a randomized response (RR) enables a function to satisfy $(\varepsilon, 0)$-DP (or ε-DP for brevity). Notably, DP is featured by the sequential composition, parallel composition, and post-processing properties. More details can be found in the Supplementary Material.

4. Our Solution: DPGEN

The detailed description of DPGEN is shown in Algorithm 1 in the Supplementary Material. In the following, we present a simplified version of DPGEN.



(a) Given a random initial image $x^{(0)}$, one uses the direction $\nabla_x \log q_\theta(x^{(t)})$ to move forward, and obtains the sampled image $x^{(T)}$ after updating T steps.

(b) Given a well-trained NN, the direction of movement required by Langevin MCMC to generate images $x^{(T)}$ can be predicted by the NN.

Figure 2. Procedures for generating images via Langevin MCMC.

4.1. Generating Images via Langevin MCMC

Recall that DPGAN can be seen as a combination of GANs and DPSGD, and the generator G_φ in DPGAN acts as a publishable image sampler to generate images. By contrast, in our DPGEN, we choose to use Langevin MCMC as an image sampler to generate the images via

$$x^{(t+1)} \leftarrow x^{(t)} + \frac{\xi^2}{2}\nabla_x \log q_\theta\left(x^{(t)}\right) + \xi z^{(t)},\ t = 0, \ldots, T-1, \quad (3)$$

where ξ denotes the step size. As the initial image $x^{(0)}$ is given randomly by the user and is independent of the training data, our contribution is to propose a DP mechanism to privatize the implied energy function $\nabla_x \log q_\theta(x)$, which provides the direction of the movement toward a perceptually realistic image in MCMC. Here, $q_\theta(x) = e^{-U_\theta(x)}/Z_\theta$ is the parameterized energy distribution, the noise $z^{(t)} \sim \mathcal{N}(0, \sigma^2 I)$, and the distribution of $x^{(T)}$ converges to the energy distribution $q_\theta(x)$ as the step $T \to \infty$ and step size $\xi \to 0$. Compared to GANs that need to interactively train two parameters, using Langevin MCMC to sample the image x requires only a single parameter θ to be trained. Moreover, Langevin MCMC is an efficient image sampler that does not

need to calculate the intractable Z_θ, indicated as

$$\nabla_x \log q_\theta(x) = -\nabla_x U_\theta(x) - \underbrace{\nabla_x \log Z_\theta}_{=0} = -\nabla_x U_\theta(x). \quad (4)$$

Thus, the image generation in Eq. (3) is guided by the direction $\nabla_{x^{(t)}} U_\theta(x^{(t)})$ from the parameterized energy function U_θ. Moreover, one can model the direction $\nabla_x U_\theta(x)$ via the neural network, and the Fisher divergence D_F serves as loss function to optimize the parameters θ formulated below:

$$D_F\left(p(x)\|q_\theta(x)\right) = \mathbb{E}_{x\sim p(x)}\left[\frac{1}{2}\left\|\nabla_x \log p(x) - \nabla_x \log q_\theta(x)\right\|^2\right]. \quad (5)$$

Since the first-order gradient function of log-PDF (i.e., Eq. (4)) is called *score*, Eq. (5) is also known as the score-matching method [22].

The procedure for generating images is shown in Figure 2, where the user can initialize a random image $x^{(0)}$ by setting each pixel as Gaussian noise. Subsequently, the image $x^{(T)}$ is generated by iterating T steps in the direction predicted by the neural network (NN). Note the privacy is not considered here and will be introduced in Section 4.2.

4.2. Privatizing Langevin MCMC Sampler

Consider $\{x_i\}_{i=1}^m$ as the sensitive dataset. The goal of privatizing Langevin MCMC is to enable the release model (that is, Eq. (4)) to generate images through Eq. (3), without causing privacy leakage. Our key idea is that Langevin MCMC can be related to DP through score matching. More specifically, as long as $\nabla_x \log p(x)$ in Eq. (5) can be privatized by the DP mechanism, the trained model satisfies DP according to the post-processing property of DP. We achieve the privatization of Langevin MCMC sampler through the following three steps (S1)∼(S3), as illustrated in Figure 1.

(S1) Gaussian Noise Addition $f(\cdot)$: Score matching as shown in Eq. (5) cannot be applied directly to our case because it requires the data distribution $p(x)$ to be differentiable everywhere. However, this is impractical for images because the pixel values of digital images are discrete. To alleviate this problem, we calculate the noisy images $\{\tilde{x}_i\}_{i=1}^m$ with $\tilde{x}_i = f(x_i) = x_i + z_i$, where z_i is sampled from Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$. We particularly note that such a noise addition is not for privacy. By doing so, we can derive $p(\tilde{x}|x)$ and joint probability $p(x, \tilde{x}) = p(\tilde{x}|x)\, p(x)$. Then, we follow [49] to derive the objective function as

$$D_F\left(p(\tilde{x})\|q_\theta(\tilde{x})\right) = \mathbb{E}_{p(\tilde{x})}\left[\frac{1}{2}\left\|\nabla_x \log p(\tilde{x}) - \nabla_x \log q_\theta(\tilde{x})\right\|^2\right]$$
$$= \mathbb{E}_{p(x,\tilde{x})}\left[\frac{1}{2}\left\|\nabla_x \log p(\tilde{x} \mid x) - \nabla_x \log q_\theta(\tilde{x})\right\|^2\right] + \text{ constant}. \quad (6)$$

As $p(\tilde{x}|x)$ follows a Gaussian distribution, we can derive $\nabla_x \log p(\tilde{x} \mid x) = (\tilde{x} - x)/\sigma^2$. Such a gradient $d = (\tilde{x} - x)/\sigma^2$, interpreted as the recovery direction, indicates how the noisy image $\tilde{x}$ is transformed or "moves" toward the training image x. Here, we in fact consider an NN with the output $\nabla_x \log q_\theta(\tilde{x})$ and use this NN output to guide MCMC. In this sense, we rewrite the objective function of the NN as

$$\ell(\theta;\sigma) \triangleq \frac{1}{2}\mathbb{E}_{p(x)}\mathbb{E}_{\tilde{x}\sim\mathcal{N}(x,\sigma^2 I)}\left[\left\|\frac{(\tilde{x}-x)}{\sigma^2} - \nabla_x \log q_\theta(\tilde{x})\right\|^2\right]. \quad (7)$$

In other words, Eq. (7) shows how the NN learns to move forward from the noisy image $\tilde{x}$ to the source image x.

(S2) RR Mechanism $H(\cdot)$: With the observation that the recovery direction $d = (\tilde{x} - x)/\sigma^2$ used for training the above NN may leak privacy, it must be privatized. After a proper privatization, one can ensure that the NN does not reveal the true position of x during the training. Hence, the NN can be released to the public, because it would be difficult for the adversary to determine if x was included in the training data.

We turn to consider how to privatize d. In the non-private setting, $\tilde{x}_i$ is supposed to point to x_i. However, in the private setting, with the randomized response (RR) as the privatization method $H(\cdot)$, $\tilde{x}_i$ is designed to point to one of its k nearest neighbors with certain probability; i.e., $x^r \triangleq \{x_i^r\}_{i=1}^m$ with $x_i^r = H(\tilde{x}_i)$, where x_i^r represents the RR result of $\tilde{x}_i$. In particular, $H(\tilde{x}_i)$ obeys the following formulation:

$$\Pr\left[H(\tilde{x}_i) = \omega\right] = \begin{cases} \dfrac{e^\varepsilon}{e^\varepsilon + k - 1}, & \omega = x_i \\ \dfrac{1}{e^\varepsilon + k - 1}, & \omega = x_i' \in X \setminus x_i \end{cases}, \quad (8)$$

where $X \triangleq \{x_j : \max(\tilde{x}_i - x_j)/\sigma_j \le \beta, j \in [m]\}$ and $|X| = k \ge 2$ (k items are sampled according to Lemma 5 of the Supplementary Material if $|X| > k$). k is a hyperparameter to be determined manually and we exmaine the impact of k on the visual quality and data utility in Section 5.3. Our objective function in Eq. (7) calculates the pixel-wise difference between $\tilde{x}$ and x. Though working in the original image space, the objective function in our design still has a theoretical support; it is extended from Fisher divergence in Eq. (5), and also has robustness to the noise [35]. Overall, given x_i, we can privatize x_i through $\mathcal{M}(x_i)$, where $\mathcal{M} \triangleq (H \circ f)$.

(S3) Training a Releasable NN Model with D: Let $D \triangleq \{(\tilde{x}_i, d_i^r)\}_{i=1}^m$, where $d_i^r = (\tilde{x}_i - x_i^r)/\sigma_i^2$, be a privatized dataset. We sample the training batches from D to train an NN for guiding the MCMC. In a nutshell, the NN trained from D satisfies DP because the NN training can be seen as post-processing, given that D is the RR result. Overall, DPGEN can be formally proved ε-DP in Theorem 1.

Theorem 1. *DPGEN satisfies ε-DP.*

5. Experiments

In this section, we demonstrate the capability of DPGEN for privately synthesizing high-resolution natural images.

Datasets. We conducted our experiments on image datasets, including MNIST [28], Fashion-MNIST[52],

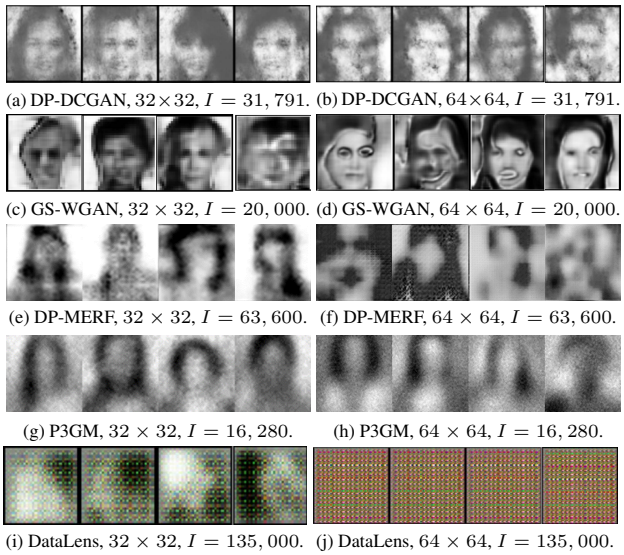
CelebA [33], and LSUN [57]. We created CelebA-Gender and CelebA-Hair datasets based on CelebA. The former is a binary classification dataset with gender as the label, while the latter is the dataset with hair color (black/blonde/brown) as the label. We created LSUN-bedroom by picking bedroom images from LSUN. All experiments were conducted using eight NVIDIA V100 GPUs, each with 32 GB RAM.

Baselines. The baseline methods in our consideration are DP-DCGAN (DCGAN trained by the built-in DPSGD in Opacus), GS-WGAN [6], DP-MERF [19], P3GM [45], DataLens [50], and G-PATE [34]. The implementation of DP-DCGAN relies on Opacus, a Facebook-supported library that enables the training of PyTorch models with DP. The implementation of the GS-WGAN, DP-MERF, DataLens, and P3GM were all based on their official source codes.

We made necessary modifications such as batch size and image size to provide experiment results in different settings. In particular, for DP-DCGAN, we only modified the training data without changing any network architecture and training parameters from Opacus. On the other hand, as the official source codes of GS-WGAN and DP-MERF can only run at a low-resolution image, we changed the network architecture (e.g., increasing the convolution channels) to acquire more patterns to improve the training stability for 32×32 and 64×64 resolutions. Due to the implementation difficulty, G-PATE results are directly excerpted from [34].

Evaluation Metrics. We demonstrate the capability of DPGEN in generating high-resolution images through (a) *visual quality*, (b) *perceptual metrics*, and (c) *downstream classification accuracy*. In particular, we first display the images of synthesized samples for visual quality comparison. Second, we evaluate the quality of synthesized samples by considering two metrics: inception score (IS) [44] and Frechet inception distance (FID) [20]. These two methods are standard in the generative model literature to evaluate the visual quality of generated images. Third, we consider a case, where we train a classifier with the synthesized samples. The testing accuracy of the classifier on real test dataset can be an indicator for the utility of the synthesized samples in the downstream classification tasks. The architecture of the classifier used in our experiment is the same as the one used in DataLens [50][1], and is shown in Figure 10 in the Supplementary Material.

Warm Start. Warm start is a technique used by DPGANs to improve their utility; this is done by initializing model parameters via pre-training them with a dataset whose distribution is similar to that of the sensitive dataset. In practice, two types of warm start method have been developed. (1) *Public Dataset:* External public dataset are preferable, if available. (2) *Dataset Partitioning:* The sensitive dataset is partitioned into two parts (e.g., the ratio of $2:98$ in [58]), where the smaller part serves to perform the pre-training. It would be difficult to obtain the corresponding external available dataset for a highly sensitive dataset in the former case. By contrast, the privacy of a certain proportion of the processed images is sacrificed in the latter case because those images are trained in their raw form. Our results are generated by DPGEN without warm start, but we will examine the impact of warm start on DPGEN in Section 5.3.



(a) DP-DCGAN, 32×32, $I = 31,791$. (b) DP-DCGAN, 64×64, $I = 31,791$.
(c) GS-WGAN, 32×32, $I = 20,000$. (d) GS-WGAN, 64×64, $I = 20,000$.
(e) DP-MERF, 32×32, $I = 63,600$. (f) DP-MERF, 64×64, $I = 63,600$.
(g) P3GM, 32×32, $I = 16,280$. (h) P3GM, 64×64, $I = 16,280$.
(i) DataLens, 32×32, $I = 135,000$. (j) DataLens, 64×64, $I = 135,000$.

Figure 3. Synthesized samples by DP-DCGAN, GS-WGAN, DP-MERF, P3GM, and DataLens on CelebA at 32×32 and 64×64 resolutions. The $\varepsilon = 10^4$ for the first four methods on CelebA (grayscale) and $\varepsilon = 10$ for DataLens on CelebA (colorful). Batch size $B = 256$. I denotes the number of training iterations.

Our Results. Below we first demonstrate the difficulty of prior work in synthesizing high-resolution natural images. Then, we show that DPGEN is capable of synthesizing natural images with resolutions of up to 128×128. Finally, we examine the influence of hyperparameters on DPGEN.

5.1. Images Synthesis from Prior Work

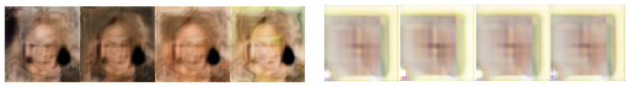
Prior Work with Large Batch Size. As the increased batch size may improve training stability, we adopted prior work to synthesize high-resolution images by significantly enlarging the batch size to up to 256. In particular, as shown in Figure 3, one can observe that even with a large ε, neither could generate facial structures to meet a decent visual quality even in the case of batch size 256. Note that the images synthesized by DataLens in Figures 3i∼3j have poor visual quality, which is slightly worse[2] than but is still roughly consistent with the results reported in [50]. From Figure 3, we demonstrate that increasing batch size (i.e., the use of more GPUs) cannot be a cure for DP complex image synthesis. Though $\varepsilon = 10^4$ implies almost zero noise, auxiliary

[1] We found the classifier architecture from the official code for [50].

[2] We derive the results in Figures 3i∼3j by running the official code. The difference might come from different number of training iterations.

steps (e.g., gradient clipping) associated with DPGANs still exist. Thus, the failure to synthesize images of acceptable visual quality in Figure 3 also provides an evidence that the baselines, in their design of network structure, cannot learn the distribution implicit in high-resolution images well.

DP Version of SOTA GANs. The 64×64 images generated by the DP version of BigGAN [3] and PGGAN [25] (termed as DP-BigGAN and DP-PGGAN) can be found in Figure 4. When implementing DP-BigGAN and DP-PGGAN, we conducted DPSGD and moments accountant [1]. We trained DP-BigGAN with $16,000$ iterations and trained DP-PGGAN with $150,000$ iterations. One can see that the images generated by DP-PGGAN and DP-BigGAN have disastrous visual quality, which justifies our claim in Section 1 that DPGANs from a simple combination of DP and SOTA GANs may be an awful design.

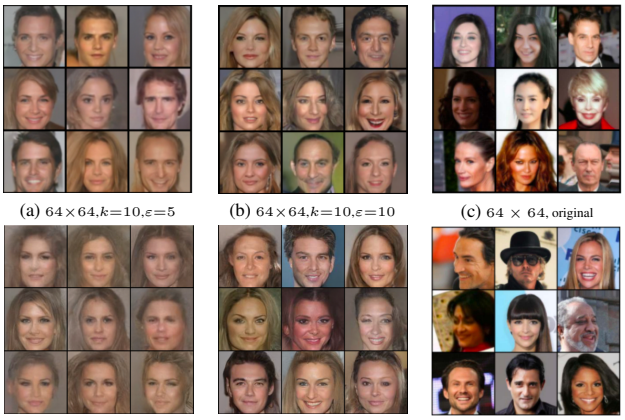


(a) DP-BigGAN with $B = 1000$ and $(7.2 \times 10^3, 10^{-5})$-DP. (b) DP-PGGAN with $B = 200$ and $(4.6 \times 10^3, 10^{-5})$-DP.

Figure 4. DP-BigGAN and DP-PGGAN. B denotes batch size.

5.2. Images Synthesis from DPGEN

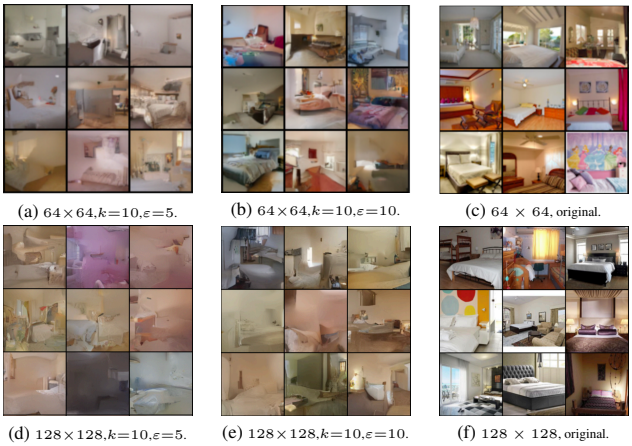
Visual Quality. We first present the visual quality evaluation results in Figure 5 and Figure 6, where all of the images were synthesized by DPGEN without warm start. Compared to the images with 64×64 resolution ($\varepsilon = 10^4$ and batch size$= 256$) in the rightmost column of Figure 3, the synthesis results of DPGEN shown in Figures 5a and 5b on images of 64×64 resolution ($\varepsilon = 5$ and 10, and batch size$= 192$) appear significantly more realistic and exhibit a much more realistic facial structure. In addition, even with the consideration of images with a resolution of 128×128, DPGEN still successfully learns the facial distribution; thus, Figures 5d and 5e show the facial structures preserved. Similar arguments apply to the experiment results for LSUN-bedroom in Figure 6. Notably, the comparisons between Figures 5a and 5b, Figures 5d and 5e, and Figures 6a and 6b show that raising ε from 5 to 10 helps preserve the color saturation. By comparing Figures 5e and 6e, we find the latter has worse quality. This can be attributed to the fact that human faces are easier to synthesize but complex scenes with various interactions among multiple objects are naturally more difficult to synthesize [13, 21]. Similar arguments apply to Figures 5a and 6a, 5b and 6b, and 5d and 6d.



(a) $64 \times 64, k=10, \varepsilon=5$ (b) $64 \times 64, k=10, \varepsilon=10$ (c) 64×64, original

(d) $128 \times 128, k=10, \varepsilon=5$ (e) $128 \times 128, k=10, \varepsilon=10$ (f) 128×128, original

Figure 5. Synthesized samples by DPGEN on the CelebA. The DPGEN results for 64×64 images are derived by training $15,000$ iterations and the DPGEN results for 128×128 images are derived by training $30,000$ iterations.



(a) $64 \times 64, k=10, \varepsilon=5$. (b) $64 \times 64, k=10, \varepsilon=10$. (c) 64×64, original.

(d) $128 \times 128, k=10, \varepsilon=5$. (e) $128 \times 128, k=10, \varepsilon=10$. (f) 128×128, original.

Figure 6. Synthesized samples by DPGEN on LSUN-bedroom. The DPGEN results for 64×64 images are derived by training $20,000$ iterations and the DPGEN results for 128×128 images are derived by training $40,000$ iterations.

Perceptual Metrics. We present the quantitative results

Distance	Resolution	CelebA				LSUN			
		ε=5	ε=10	ε=20	$\varepsilon = \infty$	ε=5	ε=10	ε=20	$\varepsilon = \infty$
IS ↑	64×64	1.3320	1.4880	1.5916	1.6591	2.3839	2.4411	2.4868	2.4891
	128×128	1.1764	1.2529	1.4503	1.5289	2.4182	2.6278	2.6553	3.4744
FID ↓	64×64	70.4802	55.9153	53.4606	50.6617	88.7134	78.9537	61.0917	43.7117
	128×128	95.8075	57.6865	55.4168	53.4558	184.1115	98.6378	83.4055	45.2868

Table 1. Perceptual scores of DPGEN with varying resolutions and ε's ($k = 10$).

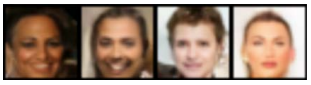
Methods	DP-DCGAN	GS-WGAN	DP-MERF	P3GM	DataLens	G-PATE	DPGEN	DPGEN
ε	10^4	10^4	10^4	10^4	10	10	10	∞
IS ↑	1.00	1.00	1.36	1.37	1.42	1.37	1.48	1.65
FID ↓	403.94	384.78	327.24	435.60	320.84	305.92	55.91	50.66

Table 2. Comparison of perceptual scores on CelebA 64×64.

Dataset \ Methods	DPGEN ($\varepsilon = \infty$)	ε	DP-DCGAN (Opacus)	GS-WGAN (NeurIPS'20)	DP-MERF (AISTATS'21)	P3GM (ICDE'21)	DataLens (CCS'21)	G-PATE (NeurIPS'21)	DPGEN (this paper)
MNIST	0.9794	$\varepsilon = 1$	0.4036	0.1432	0.6367	0.7369	0.7123	0.5880	**0.9046**
		$\varepsilon = 10$	0.8011	0.8075	0.6738	0.7981	0.8066	0.8092	**0.9357**
Fashion-MNIST	0.8794	$\varepsilon = 1$	0.1053	0.1661	0.5862	0.7223	0.6478	0.5812	**0.8283**
		$\varepsilon = 10$	0.6098	0.6579	0.6162	0.7480	0.7061	0.6934	**0.8784**
CelebA-Gender	0.8914	$\varepsilon = 1$	0.5330	0.5901	0.5936	0.5673	0.6996	0.6702	**0.6999**
		$\varepsilon = 10$	0.5211	0.6136	0.6082	0.5884	0.7287	0.6897	**0.8835**
CelebA-Hair	0.8173	$\varepsilon = 1$	0.3447	0.4203	0.4413	0.4532	0.6061	0.4985	**0.6614**
		$\varepsilon = 10$	0.3920	0.5225	0.4489	0.4858	0.6224	0.6217	**0.8147**

Table 3. Classification accuracy of the models trained on the generated data and tested on real test data under different ε's.

(a) DP-Sinkhorn, $(10, 10^{-6})$-DP.



(b) DPGEN, $k = 10, \varepsilon = 10$.

Figure 7. DP-Sinkhorn and DPGEN on CelebA (resized as 32×32).

in Table 1 to show how DPGEN behaves under different settings. In particular, for each dataset, as ε increases, we have a smaller noise scale, resulting in a higher IS (higher is better) and lower FID (lower is better). By contrast, as the resolution increases, the image complexity increases as well, resulting in a worse FID. Furthermore, in the case of $\varepsilon = 10$, the IS and FID of DPGEN synthesized images are very close to their non-private counterparts. Note that the complex structure of LSUN leads to a more diverse synthesized images, resulting in slightly higher IS in 128×128 images. As the IS's in Table 1 are quite low, a more meaningful comparison would be made between the IS in Table 1 and real IS because they are upper bounded by the real world data. The real IS for 64×64 CelebA is 2.8618, for 128×128 CelebA is 3.3020, for 64×64 LSUN is 2.6138, and for 128×128 LSUN is 3.6747. We can see that all of the IS's in Table 1 are close to the real IS.

In Table 2, we compare DPGEN with five baselines in terms of IS and FID on CelebA. Even in a more luxury setting of $\varepsilon = 10^4$ (except for DataLens with $\varepsilon = 10$), the perceptual scores of images synthesized by five baselines are inferior to those of images synthesized by DPGEN.

Classification Accuracy. Here, we make a comparison between DPGEN with ε-DP and the other baselines with (ε, δ)-DP on four different datasets under the settings of $(\varepsilon = 1, \delta = 10^{-5})$ and $(\varepsilon = 10, \delta = 10^{-5})$. One can see from Table 3 that DPGEN achieves substantially higher accuracy than all baseline methods especially when $\varepsilon = 1$. In particular, the accuracy improvement on MNIST (28×28, gray) is more than 17%. Even for a high-dimensional dataset such as CelebA-Hair (64×64, color), DPGEN still has 6% accuracy improvement, compared to the SOTA. In summary, Table 3 demonstrates the superiority of DPGEN in preserving the features in sensitive image dataset.

An Extra Comparison to DP-Sinkhorn. A very recent work, DP-Sinkhorn [5], also claims to be able to synthesize images by avoiding the training instability of GAN, similar to DPGEN. Nonetheless, despite a similar motivation, the design rationale behind DPGEN differs significantly from and still outperforms DP-Sinkhorn in terms of visual quality, as shown in Figure 7. In addition, in the case of $\varepsilon = 10$, DPGEN has FID $= 55.91$ on CelebA (resized as 64×64) but DP-Sinkhorn has FID $= 168.4$. Still in the case of $\varepsilon = 10$, DPGEN has the classification accuracy 0.884 on CelebA-Gender but DP-Sinkhorn has only classification accuracy 0.758. The above evidences reveal that DPGEN outperforms DP-Sinkhorn in terms of visual quality. Note that due to the lack of the official code, DP-Sinkhorn results are directly excerpted from [5].

5.3. Influence of Hyperparameters

	GS-WGAN	DP-MERF	P3GM	DataLens	G-PATE	DPGEN
MNIST	0.0972	0.6261	0.0820	0.2344	0.2230	**0.8194**
FMNIST	0.1000	0.5261	0.1280	0.2226	0.1874	**0.7891**

Table 4. Classification accuracy of the models under $\varepsilon = 0.2$.

Low Privacy Budget. We consider the data utility of DPGEN under the constraint of low privacy budget (higher privacy), $\varepsilon = 0.2$. Table 4 shows that DPGEN still achieves the highest accuracy compared to the other baselines, given such a tight privacy constraint. In contrast to GS-WGAN, P3GM, and DataLens, DP-MERF and DPGEN do not employ DPSGD and are the only two with a reasonable accuracy under the stringent privacy constraint. The above observation also supports our claim in Section 1 that the use

(a) Celeba (64×64), $\varepsilon = 10$ (b) Celeba (128×128), $\varepsilon = 10$

(c) LSUN-bedroom (64×64), $\varepsilon = 10$ (d) LSUN-bedroom (128×128), $\varepsilon = 10$

Figure 8. Warm start helps the image synthesis in DPGEN (2% of images are used by warm start).

Distance	Resolution	CelebA				LSUN			
		k=5	k=10	k=20	original	k=5	k=10	k=20	original
IS ↑	64×64	1.2397	1.4880	1.5795	1.6591	2.4241	2.4411	2.4754	2.4891
	128×128	1.1410	1.2529	1.2756	1.5289	2.6012	2.6278	2.6792	3.4744
FID ↓	64×64	57.3246	70.4802	95.9912	50.6617	79.1352	88.7134	141.4611	43.7117
	128×128	61.0865	90.8075	136.4811	53.4558	98.6958	204.1115	279.3343	45.2868

Table 5. Perceptual scores of DPGEN with varying resolutions and k's ($\varepsilon = 5$).

of DPSGD in image synthesis may incur too much noise, breaking the data utility. On the other hand, DPGEN has the classification accuracy 0.8194 on MNIST and 0.7891 for Fashion-MNIST in the case of $\varepsilon = 0.2$ but DP-Sinkhorn has only classification accuracy 0.832 on MNIST and 0.709 for Fashion-MNIST in the case of $\varepsilon = 10$. Thus, we believe that in the extreme case of $\varepsilon = 0.2$, DPGEN outperforms DP-Sinkhorn in terms of data utility.

Warm Start. Recall that the warm start has been widely adopted to train DPGANs, as data utility can be enhanced by sacrificing the privacy of certain images. Although we have reported the success of DP image synthesis via DPGEN in Section 5.2, we also consider whether the warm start could be applied to DPGEN, a non-DPGAN approach. To test the capability of the warm start in improving the visual quality, we trained the DPGEN both with and without warm start for $5,000$ iterations. We confirm from Figure 8 that the performance of DPGEN can also be benefited from the warm start, which helps synthesize more perceptually realistic images. Note that the results without the warm start in Figure 8a are derived by training $5,000$ iterations, while the results (without warm start) in Figure 5b are derived by training $15,000$ iterations. This explains their visual difference. Similar arguments apply to Figures 5e and 8b, Figures 6b and 8c, and Figures 6e and 8d.

Impact of k on Perceptual Metrics. The hyperparameter k controls the level of diversity in Eq. (8), which also affects the privacy-utility tradeoff. Table 5 shows how the varying k's affect the perceptual metrics. We can find that k has only mild impact on both IS and FID for the 64×64 case. We also find that IS is increased with an increased k. This can be attributed to the fact that IS is measured by the diversity of the generated images. Larger k implies more diversity and in turn better IS. Similar to our discussion on Table 1, we can see that all of the IS's in Table 5 are close to the real IS.

6. Conclusion

We propose DPGEN, a differentially private image generation method guided by an energy-based model with MCMC sampling. DPGEN is featured by its ε-DP property, in contrast to (ε, δ)-DP for nearly all the other studies. DPGEN formulates a direction toward a perceptually realistic image, which serves as a training label for the EBM, and randomizes these directions. Thus, the trained network can provide energy-guided directions with DP while generating images with the Langevin MCMC. As DPGEN is methodologically distinct in the current landscape of DP generative learning, such a design without the use of DPSGD may be a direction that deserves to be investigated. Extensive empirical experiments demonstrate that DPGEN substantially outperform the prior methods on different image datasets.

Acknowledgements. This work was supported by Ministry of Science and Technology, Taiwan, ROC, under Grants MOST 111-2636-E-A49-011 and MOST 110-2221-E-001–020-MY2. We also thank National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources.

References

[1] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *ACM Conference on Computer and Communications Security (CCS)*, 2016. 1, 2, 6

[2] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *International Conference on Machine Learning (ICML)*, 2017. 2, 3

[3] Andrew Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *International Conference on Learning Representations (ICLR)*, 2019. 6

[4] Zhiqi Bu, Sivakanth Gopi, Janardhan Kulkarni, Y. Lee, J. Shen, and U. Tantipongpipat. Fast and memory efficient differentially private-sgd via jl projections. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 2

[5] 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 (NeurIPS)*, 2021. 2, 7

[6] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. Gswgan: A gradient-sanitized approach for learning differentially private generators. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 5

[7] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Ganleaks: A taxonomy of membership inference attacks against generative models. In *ACM Conference on Computer and Communications Security (CCS)*, 2020. 1

[8] Jia-Wei Chen, Li-Ju Chen, Chia-Mu Yu, and Chun-Shien Lu. Perceptual indistinguishability-net (pi-net): Facial image obfuscation with manipulable semantics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[9] R. Chourasia, Jiayuan Ye, and R. Shokri. Differential privacy dynamics of langevin diffusion and noisy gradient descent. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 2

[10] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy-based models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 2

[11] S. Duane, A. Kennedy, B. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics Letters B*, 195:216–222, 1987. 2

[12] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014. 1, 3, 11

[13] Raghudeep Gadde, Qianli Feng, and Aleix M Martinez. Detail me more: Improving gan's photo-realism of complex scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13950–13959, 2021. 6

[14] Rinon Gal, Dana Cohen, Amit Bermano, and Daniel Cohen-Or. Swagan: A style-based wavelet-driven generative model. In *ACM SIGGRAPH*, 2021. 1

[15] Badih Ghazi, Noah Golowich, R. Kumar, Pasin Manurangsi, and Chiyuan Zhang. Randomized response with prior and applications to learning with label differential privacy. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 2

[16] Ian J. Goodfellow, Jean Pouget-Abadie, M. Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2014. 1

[17] U. Grenander and M. Miller. Representations of knowledge in complex systems. *Journal of the royal statistical society series b-methodological*, 56:549–581, 1994. 2

[18] Ishaan Gulrajani, F. Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 1

[19] 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 (AISTATS)*, 2021. 2, 5

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 5

[21] Tianyu Hua, Hongdong Zheng, Yalong Bai, Wei Zhang, Xiao-Ping Zhang, and Tao Mei. Exploiting relationship for complex-scene image generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):1584–1592, May 2021. 6

[22] A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, 2005. 4

[23] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[24] Animesh Karnewar and O. Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[25] Tero Karras, Timo Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations (ICLR)*, 2018. 1, 6

[26] Tero Karras, S. Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[27] Rithesh Kumar, Anirudh Goyal, Aaron C. Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *ArXiv: 1901.08508*, 2019. 2, 3

[28] Yann LeCun, Léon Botton, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998. 4

[29] Yann LeCun, Sumit Chopra, Raia Hadsell, Fu Jie Huang, and et al. A tutorial on energy-based learning. In *PREDICTING STRUCTURED DATA*. MIT Press, 2006. 2

[30] Jaewoo Lee and Daniel Kifer. Scaling up differentially private deep learning with fast per-example gradient clipping. *Proceedings on Privacy Enhancing Technologies (PETS)*, 2021:128 – 144, 2021. 2

[31] Guosheng Lin et al. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. *CVPR*, 2017. 15

[32] Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonthan Ullman, and Zhiwei Steven Wu. Leveraging public data for practical private query release. In *Synthetic Data Generation in conjunction with ICLR*, 2021. 2

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 5

[34] 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. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 5

[35] Siwei Lyu. Interpretation and generalization of score matching. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009. 4

[36] H. B. McMahan and G. Andrew. A general approach to adding differential privacy to iterative training procedures. *NeurIPS Workshop on Privacy Preserving Machine Learning (PPML)*, 2018. 2

[37] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018. 1

[38] Milad Nasr, Reza Shokri, and Amir Houmansadr. Improving deep learning with differential privacy using gradient encoding and denoising. In *ArXiv: 2007.11524*, 2020. 2

[39] R. Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, pages 113–162, 2010. 2

[40] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[41] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[42] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. *AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 2

[43] G. Parisi. Correlation functions and computer simulations (ii). *Nuclear Physics*, 205:337–344, 1981. 2

[44] Tim Salimans, I. Goodfellow, W. Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2016. 1, 5

[45] Shun Takagi, Tsubasa Takahashi, Yang Cao, and Masatoshi Yoshikawa. P3gm: Private high-dimensional data release via privacy preserving phased generative model. In *IEEE International Conference on Data Engineering (ICDE)*, 2021. 2, 5

[46] O. Thakkar, G. Andrew, and H. B. McMahan. Differentially private learning with adaptive clipping. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 2

[47] Reihaneh Torkzadehmahani, Peter Kairouz, and B. Paten. Dp-cgan: Differentially private synthetic data and label generation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 98–104, 2019. 1

[48] Florian Tramèr and D. Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations (ICLR)*, 2021. 2

[49] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011. 4

[50] Boxin Wang, Fan Wu, Yunhui Long, Luka Rimanic, Ce Zhang, and Bo Li. Datalens: Scalable privacy preserving training via gradient compression and aggregation. *ACM Conference on Computer and Communications Security (CCS)*, 2021. 2, 5

[51] Yu-Xiang Wang, B. Balle, and S. Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019. 2

[52] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv:1708.07747*, 2017. 4

[53] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. In *ArXiv: 1802.06739*, 2018. 2

[54] Chungui Xu, Ju Ren, Deyu Zhang, Yaoxue Zhang, Zhan Qin, and Ren Kui. Ganobfuscator: Mitigating information leakage under gan via differential privacy. *IEEE Transactions on Information Forensics and Security*, 14(9):2358–2371, 2019. 2

[55] L. Younes. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics and Stochastics Reports*, 65:177–228, 1999. 2

[56] D. Yu, Huishuai Zhang, Wei Chen, and T. Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. *International Conference on Learning Representations (ICLR)*, 2021. 2

[57] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *ArXiv:1506.03365*, 2015. 5

[58] Xinyang Zhang, Shouling Ji, and Ting Wang. Differentially private releasing via deep generative model. In *ArXiv: 1801.01594*, 2018. 2, 5

[59] Yuqing Zhu and Yu-Xiang Wang. Poission subsampled rényi differential privacy. In *International Conference on Machine Learning (ICML)*, 2019. 2