Don't Generate Me: Training Differentially Private Generative Models with Sinkhorn Divergence

Anonymous Author(s)
Affiliation
Address
email

Abstract

Although machine learning models trained on massive data have led to breakthroughs in several areas, their deployment in privacy-sensitive domains remains limited due to restricted access to data. Generative models trained with privacy constraints on private data can sidestep this challenge, providing indirect access to private data instead. We propose DP-Sinkhorn, a novel optimal transport-based generative method for learning data distributions from private data with differential privacy. DP-Sinkhorn minimizes the Sinkhorn divergence, a computationally efficient approximation to the exact optimal transport distance, between the model and data in a differentially private manner and uses a novel technique for controlling the bias-variance trade-off of gradient estimates. Unlike existing approaches for training differentially private generative models, which are mostly based on generative adversarial networks, we do not rely on adversarial objectives, which are notoriously difficult to optimize, especially in the presence of noise imposed by privacy constraints. Hence, DP-Sinkhorn is easy to train and deploy. Experimentally, we improve upon the state-of-the-art on multiple image modeling benchmarks and show differentially private synthesis of informative RGB images.

1 Introduction

Modern machine learning (ML) algorithms and their practical applications (e.g. recommender systems [1], personalized medicine [2], face recognition [3], speech synthesis [4], etc.) have become increasingly data hungry and the use of personal data is often a necessity. Consequently, the importance of privacy protection has become apparent to both the public and academia.

Differential privacy (DP) is a rigorous definition of privacy that quantifies the amount of information leaked by a user, participating in a data release [5, 6]. The degree of privacy protection is represented by the privacy budget. DP was originally designed for answering queries to statistical databases. In a typical setting, a data analyst (party wanting to use data; e.g. a healthcare company) sends a query to a data curator (party in charge of safekeeping the database; e.g. a hospital), who makes the query on the database and replies with a semi-random answer that preserves privacy. Responding to each new query incurs a privacy cost. If the analyst has multiple queries, the curator must subdivide the privacy budget to spend on each query. Once the budget is depleted, the curator can no longer respond to queries, preventing the analyst from performing new, unanticipated tasks with the database.

Generative models can be applied as a general and flexible data-sharing medium [7, 8], sidestepping the above problems. In this scenario, the curator first encodes private data into a generative model; then, the model is shared with the analyst, who can use it to synthesize similar yet different data from the training data. This data can be used in any way desired, such as for in-depth data

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

analysis or to train specific ML models. Unanticipated novel tasks can be accommodated without repeatedly interacting with the curator, since the analyst can easily generate additional synthetic data as required. So long as the generative model is learned within the privacy budget, the privacy of individuals contributing to the database remains protected.

Differentially private learning of generative models has been studied mostly using generative adversarial networks (GANs) [7, 9, 10, 11, 12, 13]. While GANs in the non-private setting can synthesize complex data such as high definition images [14, 15], their application in the private setting is challenging. This is in part because GANs suffer from training instabilities [16, 17], which can be exacerbated by adding noise to the network's gradients during training, a common technique to implement DP. Hence, GANs typically require careful hyperparameter tuning and supervision during training to avoid collapsing. This goes against the principle of privacy, where repeated interactions with data need to be avoided [18].

In this paper, we propose *DP-Sinkhorn*, a novel method to train differentially private generative models using the semi-debiased Sinkhorn loss. DP-Sinkhorn is based on the framework of optimal transport (OT), where the problem of learning a generative model is framed as minimizing the optimal transport distance, a type of Wasserstein distance, between the generator-induced distribution and the real data distribution [19, 20]. DP-Sinkhorn approximates the exact OT distance in the primal space using the Sinkhorn iteration method [21]. We also propose a novel semi-debiased Sinkhorn loss to optimally control the bias-variance trade-off when estimating gradients of this OT distance in the privacy preserving setting. Since our approach does not rely on adversarial components, it avoids any training instabilities and removes the need for early stopping. This makes our method easy to train and deploy in practice. To the best of our knowledge, DP-Sinkhorn is the first fully OT-based approach for differentially private generative modeling.

In summary, we make the following contributions: (i) We propose DP-Sinkhorn, a flexible and robust optimal transport-based framework for training differentially private generative models. (ii) We demonstrate a novel technique to finely control the bias-variance trade-off of gradient estimates when using the Sinkhorn loss. (iii) Benefiting from these technical innovations, we achieve state-of-the-art performance on widely used image modeling benchmarks for varying privacy budgets, both in terms of image quality (as measured by FID) and downstream image classification accuracy. Finally, we present informative RGB images generated under strict differential privacy without the use of public data, with image quality surpassing that of concurrent works.

2 Related Works

The task of learning generative models on private data has been tackled by many prior works. The general approach is to introduce privacy-calibrated noise into the model parameter gradients during training. While various GAN-based approaches have been introduced [7, 9, 10, 11, 12, 13], it is well documented that GANs are unstable during training [16, 17] due to the non-optimality of the discriminator producing large biases in the generator gradient [19]. This problem is critical in the context of DP, where the imposed gradient noise can increase training instabilities and where interaction with private data (e.g. hyperparameter tuning) should be limited. Our approach circumvents these issues by not relying on adversarial learning schemes. Furthermore, state-of-the-art methods [12, 13] rely on training a large number of discriminators to take advantage of the subsampling property of differential privacy. This hinders their practical usefulness as the discriminators require large amounts of GPU/TPU memory to run. In contrast, only a single generator network is trained in DP-Sinkhorn, making our approach more amenable to various hardware configurations.

Other generative models have also been studied in the DP setting. [22] partitions the private data in clusters and learns separate likelihood-based models for each cluster. [23] uses MMD with random Fourier features. While these works do not face the same stability issues as GANs, their restricted modelling capacity results in these methods mostly learning prototypes for each class. DP-Sinkhorn is better at leveraging the modeling capacity of neural networks to produce high utility synthetic data while preserving privacy. Lastly, while [24] produced strong empirical results, their privacy analysis relies on the use of Wishart noise on sample covariance matrices, which has been proven to leak privacy [25]. Hence, their privacy protection is invalid in its current form.

3 Background

3.1 Notations and Setting

Let $\mathcal{X}$ denote a sample space, $\mathcal{P}(\mathcal{X})$ all possible measures on $\mathcal{X}$, and $\mathcal{Z} \subseteq \mathbb{R}^d$ the latent space. We are interested in training a generative model $g : \mathcal{Z} \mapsto \mathcal{X}$ such that its induced distribution $\mu = g \circ \xi$ with noise source $\xi \in \mathcal{P}(\mathcal{Z})$ is similar to observed ν through an independently sampled finite sized set of observations $D = \{\mathbf{y}\}^N$. In our case, g is a trainable parametric function with parameters θ.

3.2 Generative Learning with Optimal Transport

Optimal Transport-based generative learning considers minimizing variants of the Wasserstein distance between real and generated distributions [19, 20]. Two key advantages of the Wasserstein distance over standard GANs, which optimize the Jensen-Shannon divergence [26], are its definiteness on distributions with non-overlapping supports, and its weak metrization of probability spaces [27]. This prevents collapse during training caused by discriminators that are overfit to training data.

The OT framework can be formulated in either the primal or dual formulation. In WGAN and variants [27, 28, 29], the dual potential is approximated by an adversarially trained discriminator. These methods still encounter instabilities during training, since the non-optimality of the discriminator can produce arbitrarily large biases in the generator gradient [19]. The primal formulation involves solving for the optimal transport plan—a joint distribution over the real and generated sample spaces. The distance between the two distributions is then measured as the expectation of a point-wise cost function between pairs of samples as distributed according to the transport plan.

In general, finding the optimal transport plan is a difficult optimization problem. The entropy-regularized Wasserstein distance (ERWD) imposes a strongly convex regularization term on the Wasserstein distance, making the OT problem between finite samples solvable in linear time [30]. Given a positive cost function $c : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}^+$ and $\lambda \geq 0$, the ERWD is defined as:

$$W_{c,\lambda}(\mu, \nu) = \min_{\pi \in \Pi} \int c(\mathbf{x}, \mathbf{y}) d\pi(\mathbf{x}, \mathbf{y}) + \lambda \int \log \left(\frac{d\pi(\mathbf{x}, \mathbf{y})}{d\mu(\mathbf{x}) d\nu(\mathbf{y})} \right) d\pi(\mathbf{x}, \mathbf{y}) \tag{1}$$

where $\Pi = \{\pi(\mathbf{x}, \mathbf{y}) \in \mathcal{P}(\mathcal{X} \times \mathcal{X}) | \int \pi(\mathbf{x}, \cdot) d\mathbf{x} = \nu, \int \pi(\cdot, \mathbf{y}) d\mathbf{y} = \mu\}$.

The Sinkhorn divergence uses auto-correlation terms to reduce the entropic bias introduced by ERWD with respect to the exact Wasserstein distance, canceling it out completely for $\mu = \nu$ (*i.e.* $S_{c,\lambda}(\mu, \nu) = 0$ for matching $\mu = \nu$). This results in faithful matching between the generator and real distributions. Here, we use the Sinkhorn divergence as defined in [31].

Definition 3.1. *(Sinkhorn Divergence) The Sinkhorn divergence between measures μ and ν is defined as:*

$$S_{c,\lambda}(\mu, \nu) = 2W_{c,\lambda}(\mu, \nu) - W_{c,\lambda}(\mu, \mu) - W_{c,\lambda}(\nu, \nu) \tag{2}$$

3.3 Differential Privacy

The current gold standard for measuring the privacy risk of data releasing programs is the notion of differential privacy (DP) [5]. Informally, DP measures to what degree a program's output can deviate between adjacent input datasets d and d'—sets differing by one entry. For a user contributing their data, this translates to a guarantee on how much an adversary could learn about them from observing the program's output. Here, we are learning a generative model of images, while conditioning on available semantic labels. Hence, we are interested in the domain of image-and-label datasets where each image and its label constitute an entry.

A well-studied formulation of privacy, which allows tight composition of multiple queries and can be easily converted to standard definitions of DP, is provided by Rényi Differential Privacy (RDP) [32]:

Definition 3.2. *(Rényi Differential Privacy) A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (α, ϵ)-RDP if for any adjacent $d, d' \in \mathcal{D}$:*

$$D_\alpha(\mathcal{M}(d) | \mathcal{M}(d')) \leq \epsilon, \tag{3}$$

where D_α is the Rényi divergence of order α. Also, any $\mathcal{M}$ that satisfies (α, ϵ)-RDP also satisfies $(\epsilon + \frac{\log 1/\delta}{\alpha - 1}, \delta)$-DP

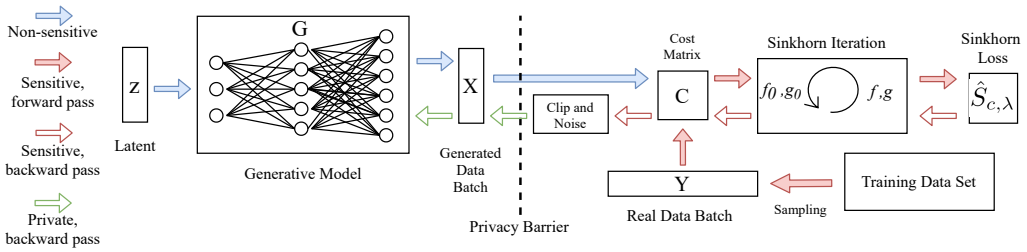


Figure 1: Flow diagram of DP-Sinkhorn for a single training iteration: Sensitive training data is combined with non-sensitive generated data in the cost matrix. Then, the loss is calculated using the Sinkhorn algorithm. In the backward pass, we impose a privacy barrier behind the generator by clipping and adding noise to the gradients at the generated image level, similar to [12].

In our case, $\mathcal{M}$ is a DP-learning algorithm, d is a training set, and $\mathcal{M}(d)$ is a generator trained on d. For sensitivity S and standard deviation of Gaussian noise σ, the Gaussian mechanism satisfies $(\alpha, \alpha S^2/(2\sigma^2))$-RDP [32]. Subsampling the dataset into batches also improves privacy. The effect of subsampling on the Gaussian mechanism under RDP has been studied in [33, 34, 35]. Privacy analysis of a gradient-based learning algorithm entails accounting for the privacy cost of single queries (possibly with subsampling), summing up the privacy cost across all queries (i.e. training iterations in our case), and then choosing the best α. A more thorough discussion of DP can be found in the Appendix.

4 Differentially Private Sinkhorn

We propose DP-Sinkhorn (Fig. 1), an OT-based method to learn differentially private generative models that avoids the training instability issues of GANs. In this section, we first provide an overview of DP-Sinkhorn, followed by our novel loss function Semi-debiased Sinkhorn loss. We then analyze the privacy protection of DP-Sinkhorn, and discuss some design considerations.

Algorithm 1 *DP-Sinkhorn*
L is number of categories. $\mathcal{X}$ is sample space. M is size of private data set. *backprop* is a reverse mode auto-differentiation function that takes 'out', 'in' and 'grad weights' as input and computes the Jacobian vector product $J_{\text{in}}(\text{out}) \cdot$ grad weights. Poisson Sample and $\hat{W}_\lambda$ (via Sinkhorn iterations) are defined in Appendix.

Input: private data set $d = \{(\mathbf{y}, \mathrm{l}) \in \mathcal{X} \times \{0, ..., L\}\}^M$, sampling ratio q, noise scale σ, clipping coefficient Δ, generator g_θ, learning rate α, entropy regularization λ, debiasing resample fraction p, total steps T.
Output: θ
$n = q * M$, $n' = floor(n * p)$
for $t = 1$ **to** T **do**
  Sample $\mathbf{Y} \leftarrow$ Poisson Sample(d, q),
  $\mathbf{Z} \leftarrow (\mathbf{z}_i)_{i=1}^{(n+n')} \overset{i.i.d.}{\sim} \text{Unif}(0,1)$
  $L_x \leftarrow \{\mathrm{l}_i\}_{i=1}^{(n+n')} \overset{i.i.d.}{\sim} \text{Unif}(0, ..., L)$
  $\mathbf{X} \leftarrow \{\mathbf{x}_i = g_\theta(\mathbf{z}_i, \mathrm{l}_i)\}_{i=1}^{(n+n')}$
  $\text{grad}_{\mathbf{X}} \leftarrow \nabla_{\mathbf{X}} \hat{S}_{c,\lambda,p}(\mathbf{X}, \mathbf{Y})$
  $\text{grad}_{\mathbf{X}[0:n]} \leftarrow clip(\text{grad}_{\mathbf{X}[0:n]}, \Delta) + 2\Delta\sigma\mathcal{N}(\vec{0}, \mathbb{I})$
  $\text{grad}_{\mathbf{X}[n:n+n']} \leftarrow clip(\text{grad}_{\mathbf{X}[n:n+n']}, \Delta)$
  $\text{grad}_\theta \leftarrow backprop(\mathbf{X}, \theta, \text{grad}_{\mathbf{X}})$
  $\theta \leftarrow \theta - \alpha * Adam(\text{grad}_\theta)$
end for

4.1 Overview of DP-Sinkhorn

DP-Sinkhorn aims to stably and robustly train generative models on high dimensional data (e.g. images) while preserving the privacy of training data. As discussed in Sec. 2, current state-of-the-art methods in privacy preserving data generation are reliant on adversarial training schemes that are not robust, unstable, and complicated to train. DP-Sinkhorn leverages advancements in OT-based generative learning to do away with the adversarial training scheme. Specifically, training a generative model with DP-Sinkhorn is an end-to-end iterative loss minimization process. In each iteration, data produced by the generator are split according to the debiasing ratio into a "cross" group and a "debiasing" group. Empirical OT distances are calculated between the "cross" group and the real data, and between the "debiasing" group and the "cross group". Gradients of the OT distances with respect to the generated data are calculated and backpropagated to the generator. Privacy protection is enforced by clipping and adding noise to the gradients of the "cross" group during backpropagation.

4.2 Estimating Sinkhorn Divergence with Semi-Debiased Sinkhorn Loss

Sinkhorn divergence, as expressed in Eq. 2, involves integration over the sample space. Empirical estimation of Eq. 2 based on finite samples is required to train a generative model through gradient-based optimization. An obvious solution suggested by previous works [31, 20] would be to replace μ and ν with empirical samples from each distribution.

Definition 4.1. *The empirical Sinkhorn loss computed over a batch of n generated examples $\mathbf{X}$ and m real examples $\mathbf{Y}$ is defined as [31]:*

$$\hat{S}_{c,\lambda}(\mathbf{X},\mathbf{Y}) = 2\hat{W}_\lambda(\mathbf{X},\mathbf{Y}) - \hat{W}_\lambda(\mathbf{X},\mathbf{X}) - \hat{W}_\lambda(\mathbf{Y},\mathbf{Y}), \tag{4}$$

where $\hat{W}_\lambda(\mathbf{A},\mathbf{B}) = C_{\mathbf{AB}} \odot P^\lambda_{\mathbf{AB}}$.[1] $C_{\mathbf{AB}} \in \mathbb{R}^{+^{n\times m}}$ is the cost matrix between $\mathbf{A}$ and $\mathbf{B}$, and $P^\lambda_{\mathbf{AB}}$ is the approximate optimal transport plan that empirically minimizes $\hat{W}_\lambda(\mathbf{A},\mathbf{B})$ computed by the Sinkhorn algorithm.

However, [36] showed that the gradients of $\hat{S}_{c,\lambda}(\mathbf{X},\mathbf{Y})$ are biased estimates of the gradients of $S_{c,\lambda}(\mu,\nu)$, computed over the population. Instead, they proposed a loss formulation that produces unbiased gradients using additional independently drawn samples:

Definition 4.2. *Following the notations of Def. 4.1, the debiased Sinkhorn loss is defined as [36]:*

$$\hat{S}_{c,\lambda}(\mathbf{X},\mathbf{Y},\mathbf{X}',\mathbf{Y}') = 2\hat{W}_\lambda(\mathbf{X},\mathbf{Y}) - \hat{W}_\lambda(\mathbf{X},\mathbf{X}') - \hat{W}_\lambda(\mathbf{Y},\mathbf{Y}'). \tag{5}$$

In comparison with Def. 4.1, Def. 4.2 comes with higher variance (only $\hat{W}_\lambda(\mathbf{X},\mathbf{Y})$ contributes to variance in Def. 4.1). Unfortunately, privacy constraints in the DP setting prevent us from using very large batch sizes or very long training periods with low learning rates to effectively reduce variance. Hence, the variance incurred from using the unbiased estimator is more difficult to handle in the DP setup. Furthermore, Def. 4.2 draws two batches of real data in every training step, thereby increasing the privacy cost of each step. Nonetheless, Def. 4.2 is an unbiased estimator with better convergence properties. We now discuss how we overcome the above issues in DP-Sinkhorn.

First, we make the observation that the $\hat{W}_\lambda(\mathbf{Y},\mathbf{Y}')$ term does not contribute to gradients of the generator. Hence, we can omit it from $\hat{S}_{c,\lambda}$. Next, we propose a loss formulation that interpolates between biased and unbiased Sinkhorn divergence. As shown in previous works, it can be beneficial to control the bias-variance trade-off through mixing biased and unbiased gradient estimators [37]. Instead of completely resampling the generator for $\mathbf{X}'$, we reuse some of the samples in $\mathbf{X}$ when computing $\hat{W}_\lambda(\mathbf{X},\mathbf{X}')$. This provides better control over the bias-variance trade-off when empirically estimating gradients.

Definition 4.3. *(Semi-debiased Sinkhorn loss) For a mixture fraction $p \in [0,1]$ and natural number n, $n' = floor(n \times p)$. Given $n+n'$ generated samples $\mathbf{X} \in \mathcal{X}^{n+n'}$ and m real samples $\mathbf{Y} \in \mathcal{X}^m$, the semi-debiased Sinkhorn loss is defined as:*

$$\hat{S}_{c,\lambda,p}(\mathbf{X},\mathbf{Y}) = 2\hat{W}_\lambda(\mathbf{X}^{[0:n]},\mathbf{Y}) - \hat{W}_\lambda(\mathbf{X}^{[0:n]},\mathbf{X}^{[n':n+n']}), \tag{6}$$

where $\mathbf{X}^{[a:b]}$ denotes the contiguous rows of $\mathbf{X}$ starting from a and ending with $b-1$.

When $p=1$, Eq. 6 is equal to Eq. 5, whereas when $p=0$, Eq. 6 recovers Eq. 4 (ignoring the terms in Eqs. 4 and 5 that only depend on data $\mathbf{Y}$ and are irrelevant during training).

Algorithm 1 describes how Eq. 6 is used to train a generative model, while additionally modifying the gradient by adding noise and clipping to implement the privacy mechanism described below. Training of the generator proceeds by computing the gradient of the semi-debiased Sinkhorn loss with respect to $\mathbf{X}$. Please also see the Appendix for more details.

4.3 Privacy Protection

Information about real data enters the generator through loss function gradients with respect to the generated images. Let $\mathbf{G} = \nabla_{\mathbf{X}}\hat{S}_{c,\lambda,p}(\mathbf{X},\mathbf{Y})$ denote the gradients of the semi-debiased Sinkhorn

[1] $\odot$ is the Hadamard product

loss, and let $\mathbf{G}^{[a:b]}$ denote the contiguous rows of $\mathbf{G}$ from a to $b-1$ inclusive. We modify $\mathbf{G}$ as:

$$\tilde{\mathbf{G}} = \mathbf{G}^{[0:n]} \cdot \min\left(\frac{\Delta}{||\mathbf{G}^{[0:n]}||_2}, 1\right), \quad \tilde{\mathbf{G}}' = \mathbf{G}^{[n:n+n']} \cdot \min\left(\frac{\Delta}{||\mathbf{G}^{[n:n+n']}||_2}, 1\right)$$

$$\hat{\mathbf{G}} = \text{concat}(\tilde{\mathbf{G}} + \gamma, \tilde{\mathbf{G}}'), \quad \text{where } \gamma \sim \mathcal{N}(0, \Delta^2\sigma^2), \quad \text{concat is applied to the first axis}$$

We observe that $\nabla_{\mathbf{X}^{[n:n+n']}} \hat{W}_\lambda(\mathbf{X}^{[0:n]}, \mathbf{Y}) = 0$, i.e. $\mathbf{G}^{[n:n+n']}$ contains no information about $\mathbf{Y}$. As such, noise need not be added to this term, but we apply clipping to $\mathbf{G}^{[n:n+n']}$ to preserve the scale between the magnitudes of the gradients. The following theorem states the privacy guarantee of DP-Sinkhorn's gradient updates, with proofs in the appendix:

Theorem 4.1. *For clipping constant Δ and noise vector $\gamma \sim \mathcal{N}(0, \Delta^2\sigma^2)$, releasing $\hat{\mathbf{G}}$ satisfies $(\alpha, 2\alpha/\sigma^2)$-RDP.*

We use the RDP accountant with Poisson subsampling proposed in [35] for privacy composition across updates. Note that the batch size of $\mathbf{X}$ is kept fixed, while the batch size of $\mathbf{Y}$ follows a binomial distribution due to Poisson subsampling.

4.4 Design Considerations

Advantages of primal form OT When compared to WGAN, learning with primal form OT (such as Sinkhorn divergence) has distinct differences. While both are approximations to the exact Wasserstein distance, the source of the approximation error differs. WGAN's source of error lies in the sub-optimality of the dual potential function. Since this potential function is parameterized by an adversarially trained deep neural network, it enjoys neither convergence guarantees nor feasibility guarantees. Furthermore, the adversarial training scheme can cause the discriminator and generator to change abruptly every iteration to counter the strategy of the other player from the previous iteration [38], resulting in non-convergence. In contrast, the suboptimality of the transport plan when computing Sinkhorn divergence can be controlled by using enough iterations, and the bias introduced by entropic regularization can be controlled by using small values of λ. Training with the Sinkhorn divergence does not involve any adversarial training at all, converges more stably, and reaps the benefits of OT metrics at covering modes.

Cost function The choice of the element-wise cost $c(\mathbf{x}, \mathbf{y})$ influences the type of images produced by the generator. We consider a mixture between pixel-wise L_1 and squared L_2 losses. L_2 loss has smooth gradients that scale with the difference in pixel value, whereas the gradient of L_1 loss is constant in magnitude for each pixel that differs. Therefore, while L_2 loss can quickly rein in outlier pixel values, L_1 loss can encourage generated image pixels to closely match those of the real image, promoting sharpness. We define the element-wise cost function as $c_m(\mathbf{x}, \mathbf{y}) = L_2(\mathbf{x}, \mathbf{y})^2 + m\, L_1(\mathbf{x}, \mathbf{y})$, where $L_2(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2$, $L_1(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}|$ and m is a scalar mixture weight. Class conditioning is also achieved through the cost function by concatenating an one-hot class embedding to both the generated images and real images, similar to the approach used in [36]. Intuitively, this works by increasing the cost between image pairs of different classes, hence shifting the weight of the transport plan (P^*_λ in Eq. 4) towards class-matched pairs.

5 Experiments

We conduct experiments on differentially private conditional image synthesis, since our focus is on generating high-dimensional data with privacy protection. We evaluate our method on both visual quality and data utility for downstream classification tasks. Additional experiments and analyses of the proposed semi-debiased Sinkhorn loss can be found in the Appendix.

5.1 Experimental Setup

Datasets We use 3 image datasets: MNIST [40], Fashion-MNIST [41], and CelebA [42] downsampled to 32x32 pixels. For MNIST and Fashion-MNIST, generation is conditioned on regular class labels; for CelebA we condition on gender.

Metrics In all experiments, we compute metrics against a synthetic dataset of 60k image-label pairs sampled from the model. For a quantitative measure of visual quality, we report FID [43]. To measure the utility of generated data, we assess the class prediction accuracy of classifiers trained with synthetic data on the real test sets. We consider logistic regression, MLP, and CNN classifiers.

Table 1: Comparison of DP image generation results on MNIST and Fashion-MNIST at $(\epsilon, \delta) = (10, 10^{-5})$-DP. Results for other methods (G-PATE [39], DP-MERF AE [23], DP-CGAN [9], GS-WGAN [12]) are from [12], except Datalense [13]. Results averaged over 5 runs of synthetic dataset generation and classifier training.

Method	DP-ϵ	MNIST				Fashion-MNIST			
		FID	Acc (%)			FID	Acc (%)		
			Log Reg	MLP	CNN		Log Reg	MLP	CNN
Real data	∞	1.6	92.2	97.5	99.3	2.5	84.5	88.2	90.8
Non-priv Sinkhorn ($m = 1$)	∞	54.2	89.0	89.0	91.0	65.8	78.4	79.1	73.9
Non-priv Sinkhorn ($m = 3$)	∞	43.4	87.7	87.3	90.6	63.8	78.4	78.4	73.3
G-PATE	10	177.2	26	25	51/80.9[3]	205.8	42	30	50/69.3[3]
DP-CGAN	10	179.2	60	60	63	243.8	51	50	46
DP-MERF AE	10	161.1	54	55	68	213.6	50	56	62
DataLens	10	173.5	N/A	N/A	80.66	167.7	N/A	N/A	**70.61**
GS-WGAN	10	61.3	79	79	80	131.3	68	65	65
DP-Sinkhorn ($m = 1$)	10	61.2	**79.5**	**80.2**	**83.2**	145.1	**73.0**	**72.8**	69.3
DP-Sinkhorn ($m = 3$)	10	**55.56**	79.1	79.2	79.1	**129.4**	70.2	70.2	64.2

Architectures & Hyperparameters We implement DP-Sinkhorn with two generator architectures. We adopt a four layer, convolutional architecture from DCGAN [44] for MNIST and Fashion-MNIST experiments, and a twelve layer residual architecture from BigGAN [14] for CelebA experiments. Class conditioning is achieved by providing a one-hot encoding of the label to the generator, and concatenating the one-hot encoding to images when calculating the element-wise cost. We set λ=0.05 for MNIST and Fashion-MNIST experiments, and λ=5 for CelebA experiments. Complete implementation details can be found in the Appendix.

Privacy Implementation Our models are implemented in PyTorch. We implement the gradient sanitization mechanism by registering a backward hook to the generator output. MNIST and Fashion-MNIST experiments target $(10, 10^{-5})$-DP while CelebA experiments target $(10, 10^{-6})$-DP. Details are in the Appendix.

5.2 Experimental Results on Standard Benchmarks

In Table 1, we compare the performance of two DP-Sinkhorn variants with other methods on MNIST and Fashion-MNIST. We use p=0.2 for the semi-debiased loss, which was determined through grid search. The two variants use different weights (m=1 and m=3) for the L_1 loss in the cost function. Given the same privacy budget, DP-Sinkhorn with m=1 generates more informative examples than previous methods, as demonstrated by the higher accuracy achieved by the downstream classifier. On the more visually complex Fashion-MNIST, DP-Sinkhorn's lead is especially pronounced, beating previous state-of-the-art results by a significant margin. DP-Sinkhorn with m=3 achieves lower FID than all baselines, while still maintaining downstream accuracy similar to GS-WGAN. We hypothesize that giving more weight to the L_1 loss improves FID because L_1 is more sensitive to small differences in pixel values, thereby encouraging sharper edges. Images generated by DP-Sinkhorn are visualized in Fig. 2. DP-Sinkhorn produces more visual diversity within each class compared to the baselines, which likely benefits DP-Sinkhorn's downstream classification performance.

Robustness We evaluate the training stability of DP-Sinkhorn ($m = 1$, $p = 0.2$) with different learning rates and two optimizers (Adam [45] and SGD) on MNIST. We perform the same parameter sweep on GS-WGAN for comparison[2], as it is the strongest baseline we are comparing to. Results are illustrated in Fig. 4a. We find that DP-Sinkhorn reliably converges for sufficiently small learning rates, and it is not sensitive to the choice of optimizer. In contrast, GS-WGAN, relying on adversarial training, suffers from non-convergence for learning rates too big or too small, and is very sensitive to the choice of optimizer. Exact numbers are reported in the Appendix.

Privacy Utility Trade-off Stronger privacy protection can be attained by training DP-Sinkhorn for fewer iterations at the cost of utility and image quality. We evaluate the performance of DP-Sinkhorn at various privacy budgets and contrast it to GS-WGAN (Fig. 4b). DP-Sinkhorn converges quicker than GS-WGAN, shows strong performance among a wide range of privacy budgets, and provides

[3]The G-PATE authors report much more accurate classification results than reported in [12]. The visual quality of samples in both papers is roughly the same.

[2]`https://github.com/DingfanChen/GS-WGAN` (MIT License) [12]

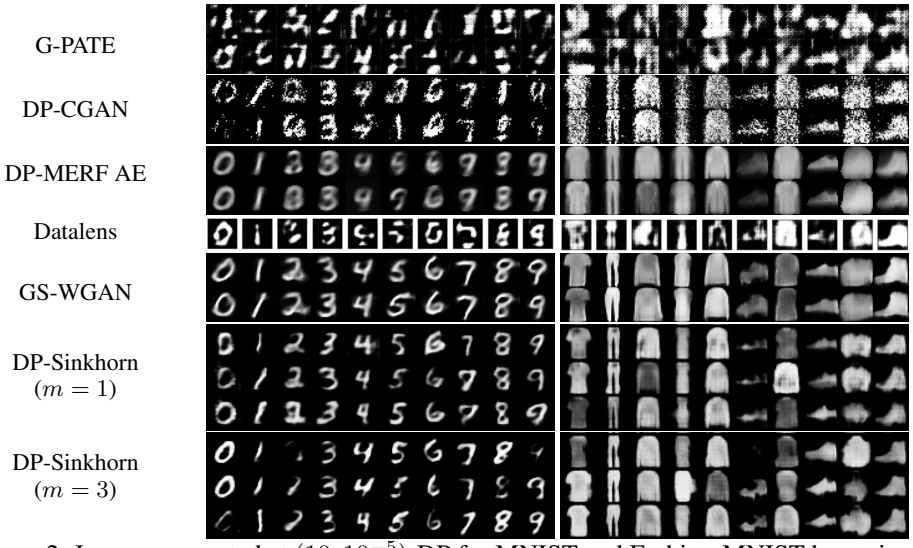


Figure 2: Images generated at $(10, 10^{-5})$-DP for MNIST and Fashion-MNIST by various methods. Datalens images obtained from [13]; images of other methods obtained from [12].

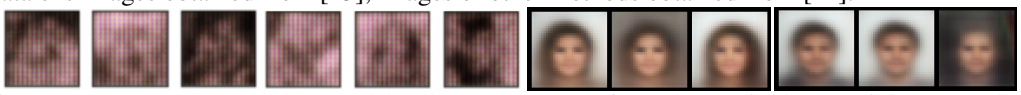


Figure 3: Images generated on CelebA by Datalens (Left) and DP-Sinkhorn (Right). Datalens images obtained from [13].

good downstream utility even at a small privacy budget of $\epsilon = 2.33$, significantly outperforming GS-WGAN. Note that we found GS-WGAN to require significantly more memory than DP-Sinkhorn, since it uses multiple discriminators for different parts of the data. In our experiments, DP-Sinkhorn can fit comfortably on an 11GB GPU, while GS-WGAN requires 24GB of GPU memory. Hence, DP-Sinkhorn is arguably more scalable to very large datsets.

Analysis of Semi-debiased Sinkhorn Loss To study why our novel semi-debiased Sinkhorn loss outperforms both fully debiased and fully biased Sinkhorn losses, we evaluate bias and variance of the semi-debiased Sinkhorn loss-based gradient estimator. We sample generator gradients with respect to the semi-debiased Sinkhorn loss with different p and plot bias and variance (Fig. 4c). Each line represents a generator trained with a different p on MNIST. Variances of each model's gradients are normalized with respect to variance at p=0. As expected, variance grows and bias decreases, as p increases. The unmitigated increase in variance at high p is detrimental to performance in the DP setup. Semi-debiasing provides control over the trade-off between consistent low-variance gradients and less biased objectives, with optimal p=0.2 in between 0 and 1, as our grid search determined.

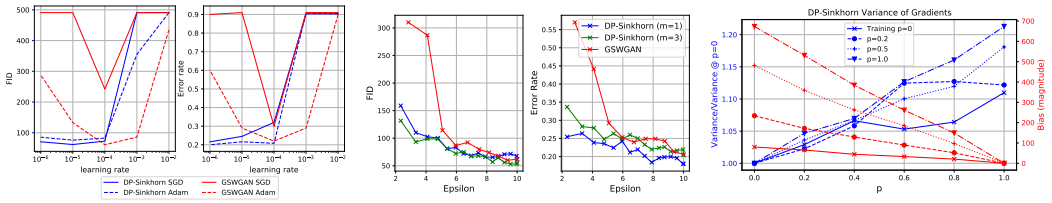
Table 2: Ablating loss functions, debiasing, and gradient perturbation mechanism on MNIST.

Image Gradient Perturbation	Loss	Debiasing	FID	Acc (%)	
				MLP	CNN
No	L_2	No	218.6	79.9	80.7
Yes	L_2	No	124.3	82.0	80.8
Yes	L_1	No	73.9	68.4	65.7
Yes	L_1+L_2	No	88.6	76.6	76.1
Yes	L_1+L_2	Full	98.0	63.0	60.5
Yes	L_1+L_2	Semi	61.2	80.2	83.2

Table 3: DP image generation results on downsampled CelebA. We include results from [13] for context, but note that their experiment uses a 64x64 resolution and a larger δ of 10^{-5}

Method	$(\epsilon, 10^{-6})$-DP	FID	Acc (%)	
			MLP	CNN
Real data	∞	1.1	91.9	95.0
Sinkhorn	∞	129.5	80.8	82.2
DP-Sinkhorn	10	168.4	76.2	75.8
DataLens [13]	$(10, 10^{-5})$	320.8	N/A	72.9

Ablations We study the impact of perturbing image vs. parameter gradients, design of element-wise cost function, and debiasing on performance in the MNIST benchmark. We start with the simplest model, using parameter gradient perturbation, L_2 loss and no debiasing, and incrementally add components. We use m=1 when adding L_1 loss, and p=0.2 when adding semi-debiasing. The clipping bound Δ is tuned separately for the variant with parameter gradient perturbation, while the other hyperparameters are kept fixed. In Table 2, we see that DP-Sinkhorn with parameter gradients is already competitive in downstream accuracy, but has poor FID in comparison to using image gradients. We observe that DP-Sinkhorn with L_2 loss yields good downsteam task performance, but has



(a) Comparing hyperparameter sensitivity of DP-Sinkhorn to GS-WGAN on MNIST. Error rate is calculated as $1 -$ Accuracy

(b) FID and utility of DP-Sinkhorn (m=1 and m=3), and GS-WGAN at various ϵ on MNIST.

(c) Bias-variance trade-off of the gradient estimator over semi-debiasing parameter p.

Figure 4: Analyzing hyperparameter choices in DP-Sinkhorn.

higher FID than the L_1 loss variant. Mixing L_1 and L_2 loss strikes a balance between better FID and downstream accuracy. We also observe that using a fully debiased gradient estimator is detrimental to performance, which we postulate is due to its high variance. The semi-debiased variant performs better than both the biased and the debiased variants.

5.3 Experimental Results on CelebA

We also evaluate DP-Sinkhorn on downsampled CelebA. We evaluate whether DP-Sinkhorn is able to synthesize RGB images that are informative for downstream classification. Despite its simplicity, DP-Sinkhorn generates informative images for gender classification, as seen in Tab. 3 (uninformative images would correspond to a $\approx 50\%$ classification ratio). Qualitatively, Fig. 3 shows that DP-Sinkhorn can learn meaningful representations of each semantic class (male and female) and produces some in-class variations, while avoiding details that could uniquely identify individuals. Concurrent to our work, Datalens[13] was also applied to gender conditioned generation of CelebA images, albeit with a different image resolution than ours. Images generated by DP-Sinkhorn clearly resemble faces, while those generated by Datalens are blurrier. We also attempted to train GS-WGAN on CelebA, but couldn't obtain meaningful results using the default hyper-parameters.

6 Conclusions

We propose DP-Sinkhorn, a novel optimal transport-based differentially private generative model. Our approach minimizes a new semi-debiased Sinkhorn loss in a differentially private manner. It does not require any adversarial techniques that are challenging to optimize. Consequently, DP-Sinkhorn is easy to train, which we hope will help its adoption in practice. We experimentally demonstrate superior performance compared to the previous state-of-the-art both in terms of image quality and on standard image classification benchmarks using data generated under DP. Our model is applicable for varying privacy budgets and is capable of synthesizing informative RGB images in a differentially private way without using additional public data. We conclude that robust models such as ours are a promising direction for differentially private generative modeling.

Limitations Our main experiments only used simple pixel-wise L_1- and L_2-losses as cost function, yet achieve better performance than GAN based methods. This suggests that in the DP setting complexity in model and objective are not necessarily beneficial. Nonetheless, limited image quality is the main challenge in DP generative modeling and future work includes designing more expressive generator networks that can further improve synthesis quality, while satisfying differential privacy. To this end, kernel-based cost functions may provide better performance on suitable datasets.

Broader impact Our work improves the state-of-the-art in privacy preserving generative modeling. Such advances promise significant benefits to the machine learning community, by allowing sensitive data to be shared more broadly via privacy preserving generative models. We believe the strong performance and robustness of DP-Sinkhorn will facilitate its adoption by practitioners. Although DP-Sinkhorn provides privacy protection in generative learning, information about individuals cannot be eliminated entirely, as no useful model can be learned under $(0, 0)$-DP. This should be communicated clearly to dataset participants. We recognize that classifiers learned with DP can potentially underperform for minority members within the dataset [46, 47, 48], which may also be the case for classifiers trained on data produced by DP-Sinkhorn. Addressing these types of imbalances is an active area of research [49, 50, 51, 52].

References

[1] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, Dec. 2016.

[2] D. Ho, S. R. Quake, E. R. B. McCabe, W. J. Chng, E. K. Chow, X. Ding, B. D. Gelb, G. S. Ginsburg, J. Hassenstab, C.-M. Ho, W. C. Mobley, G. P. Nolan, S. T. Rosen, P. Tan, Y. Yen, and A. Zarrinpar, "Enabling Technologies for Personalized and Precision Medicine," *Trends Biotechnol.*, vol. 38, no. 5, pp. 497–518, 2020.

[3] M. Wang and W. Deng, "Deep Face Recognition: A Survey," *arXiv preprint arXiv:1804.06655*, 2020.

[4] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*, pp. 265–284, Springer, 2006.

[6] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, p. 211–407, Aug. 2014.

[7] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *arXiv preprint arXiv:1802.06739*, 2018.

[8] S. Augenstein, H. B. McMahan, D. Ramage, S. Ramaswamy, P. Kairouz, M. Chen, R. Mathews, and B. A. y Arcas, "Generative Models for Effective ML on Private, Decentralized Datasets," in *International Conference on Learning Representations*, 2020.

[9] R. Torkzadehmahani, P. Kairouz, and B. Paten, "Dp-cgan: Differentially private synthetic data and label generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[10] L. Frigerio, A. S. de Oliveira, L. Gomez, and P. Duverger, "Differentially private generative adversarial networks for time series, continuous, and discrete open data," in *IFIP International Conference on ICT Systems Security and Privacy Protection*, pp. 151–164, Springer, 2019.

[11] J. Yoon, J. Jordon, and M. van der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in *International Conference on Learning Representations*, 2019.

[12] D. Chen, T. Orekondy, and M. Fritz, "GS-WGAN: A Gradient-Sanitized Approach for Learning Differentially Private Generators," in *Advances in Neural Information Processing Systems*, 2020.

[13] B. Wang, F. Wu, Y. Long, L. Rimanic, C. Zhang, and B. Li, "Datalens: Scalable privacy preserving training via gradient compression and aggregation," *arXiv preprint arXiv:2103.11109*, 2021.

[14] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2019.

[15] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

[16] M. Arjovsky and L. Bottou, "Towards Principled Methods for Training Generative Adversarial Networks," in *International Conference on Learning Representations*, 2017.

[17] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for GANs do actually converge?," vol. 80 of *Proceedings of Machine Learning Research*, (Stockholmsmässan, Stockholm Sweden), pp. 3481–3490, PMLR, 10–15 Jul 2018.

[18] K. Chaudhuri and S. A. Vinterbo, "A stability-based validation procedure for differentially private machine learning," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 2652–2660, Curran Associates, Inc., 2013.

[19] O. Bousquet, S. Gelly, I. Tolstikhin, C.-J. Simon-Gabriel, and B. Schoelkopf, "From optimal transport to generative modeling: the vegan cookbook," *arXiv preprint arXiv:1705.07642*, 2017.

[20] G. Peyré and M. Cuturi, "Computational Optimal Transport," *Foundations and Trends in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[21] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in neural information processing systems*, pp. 2292–2300, 2013.

[22] G. Acs, L. Melis, C. Castelluccia, and E. De Cristofaro, "Differentially private mixture of generative neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 6, pp. 1109–1121, 2018.

[23] F. Harder, K. Adamczewski, and M. Park, "Differentially private mean embeddings with random features (dp-merf) for simple & practical synthetic data generation," *arXiv preprint arXiv:2002.11603*, 2020.

[24] S. Takagi, T. Takahashi, Y. Cao, and M. Yoshikawa, "P3gm: Private high-dimensional data release via privacy preserving phased generative model," *arXiv preprint arXiv:2006.12101*, 2020.

[25] A. Sarwate, "Retraction for symmetric matrix perturbation for differentially-private principal component analysis," 2017.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

[27] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, pp. 5767–5777, 2017.

[29] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018.

[30] G. Peyré, M. Cuturi, *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[31] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouvé, and G. Peyré, "Interpolating between optimal transport and mmd using sinkhorn divergences," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2681–2690, 2019.

[32] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, 2017.

[33] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled rényi differential privacy and analytical moments accountant," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1226–1235, PMLR, 2019.

[34] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: Tight analyses via couplings and divergences," in *Advances in Neural Information Processing Systems*, pp. 6277–6287, 2018.

[35] Y. Zhu and Y.-X. Wang, "Poission subsampled rényi differential privacy," in *International Conference on Machine Learning*, pp. 7634–7642, 2019.

[36] T. Salimans, H. Zhang, A. Radford, and D. Metaxas, "Improving GANs using optimal transport," in *International Conference on Learning Representations*, 2018.

[37] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *International Conference on Machine Learning*, pp. 5171–5180, PMLR, 2019.

[38] L. Mescheder, S. Nowozin, and A. Geiger, "The numerics of gans," in *Advances in Neural Information Processing Systems*, pp. 1825–1835, 2017.

[39] Y. Long, S. Lin, Z. Yang, C. A. Gunter, H. Liu, and B. Li, "Scalable differentially private data generation via private aggregation of teacher ensembles," 2019.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[41] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[42] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[43] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in neural information processing systems*, pp. 6626–6637, 2017.

[44] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[46] R. Cummings, V. Gupta, D. Kimpara, and J. Morgenstern, "On the compatibility of privacy and fairness," in *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, UMAP'19 Adjunct, (New York, NY, USA), p. 309–315, Association for Computing Machinery, 2019.

[47] S. Kuppam, R. McKenna, D. Pujol, M. Hay, A. Machanavajjhala, and G. Miklau, "Fair decision making using privacy-protected data," *CoRR*, vol. abs/1905.12744, 2019.

[48] S. Agarwal. PhD thesis, University of Waterloo, 2020.

[49] A. Grover, J. Song, A. Kapoor, K. Tran, A. Agarwal, E. J. Horvitz, and S. Ermon, "Bias correction of learned generative models using likelihood-free importance weighting," in *Advances in Neural Information Processing Systems*, 2019.

[50] K. Choi, A. Grover, T. Singh, R. Shu, and S. Ermon, "Fair generative modeling via weak supervision," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[51] N. Yu, K. Li, P. Zhou, J. Malik, L. Davis, and M. Fritz, "Inclusive GAN: improving data and minority coverage in generative models," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXII*, 2020.

[52] J. Lee, H. Kim, Y. Hong, and H. W. Chung, "Self-diagnosing gan: Diagnosing underrepresented samples in generative adversarial networks," *arXiv preprint arXiv:2102.12033*, 2021.

[53] J. Feydy. PhD thesis, ENS, Mar 2020.

[54] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, (New York, NY, USA), p. 308–318, Association for Computing Machinery, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Sec. 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Sec. 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We did not derive any theoretical results but stated assumptions of those we've referenced and used.
   (b) Did you include complete proofs of all theoretical results? [Yes] See appendix for proof of Theorem 4.1.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We will release code and instructions to reproduce the main experiments upon acceptance of the manuscript. The internal guidelines of our institution prevent us from releasing code at this stage.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Standard splits are used. Hyperparameters are provided in Section 5 and the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We reported average of 5 repeated runs. Error bars/ranges are not included due to formatting considerations and following conventions from previous works.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We show results and images from previous, publicly available papers, which we cite, and we run publicly available code for baseline comparisons, similarly citing the work (GS-WGAN [12]). Software packages used in our experiments are cited and described in Appendix D.
   (b) Did you mention the license of the assets? [Yes] We point out that the code we used (`https://github.com/DingfanChen/GS-WGAN`) to run baseline experiments was released under the MIT license (see footnote in Sec. 5.2).
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We used publically available data in our experiments. We did not curate new datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We did not curate new datasets. Existing datasets that we used are described in Appendix D.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not conduct human experiments or use crowdsourcing.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]