Pre-trained Perceptual Features Improve Differentially Private Image Generation

Anonymous authors
Paper under double-blind review

Abstract

Training even moderately-sized generative models with differentially-private stochastic gradient descent (DP-SGD) is difficult: the required level of noise for reasonable levels of privacy is simply too large. We advocate instead building off a good, relevant representation on an informative public dataset, then learning to model the private data with that representation. In particular, we minimize the maximum mean discrepancy (MMD) between private target data and a generator's distribution, using a kernel based on perceptual features learned from a public dataset. With the MMD, we can simply privatize the data-dependent term once and for all, rather than introducing noise at each step of optimization as in DP-SGD. Our algorithm allows us to generate CIFAR10-level images with $\epsilon \approx 2$ which capture distinctive features in the distribution, far surpassing the current state of the art, which mostly focuses on datasets such as MNIST and FashionMNIST at a large $\epsilon \approx 10$. Our work introduces simple yet powerful foundations for reducing the gap between private and non-private deep generative models. Our code is available at `https://anonymous.4open.science/r/dp-gfmn`.

1 INTRODUCTION

The gold standard privacy notion, *differential privacy* (DP), is now ubiquitous in a diverse range of academic research, industry products (Apple, 2017), and even government databases (National Conference of State Legislatures, 2021). DP provides a mathematically provable privacy guarantee, which is its main strength and reason for its popularity. However, one of the properties of DP is *composability*, meaning data can be accessed more than once – but the level of privacy guarantee degrades each time. To guarantee a high level of privacy, one needs to limit access to data, a challenge in applying DP with the usual iterative optimization algorithms used in machine learning.

Differentially private data generation solves this problem by creating a synthetic dataset that is *similar* to the private dataset, in terms of some chosen similarity metric. While producing such a synthetic dataset incurs a privacy loss, the resulting dataset can be used repeatedly without further loss of privacy. Classical approaches, however, typically assume a certain class of pre-specified purposes on how the synthetic data can be used (Mohammed et al., 2011; Xiao et al., 2010; Hardt et al., 2012; Zhu et al., 2017). If data analysts use the data for other tasks outside these pre-specified purposes, the theoretical guarantees on its utility are lost.

To produce synthetic data usable for potentially *any* purpose, many papers on DP data generation have utilized the recent advances in deep generative modelling. The majority of these approaches are based on the generative adversarial network (GAN; Goodfellow et al., 2014) framework, where a discriminator and a generator play an adversarial game to optimize a given distance metric between the true and synthetic data distributions. Most approaches under this framework have used DP-SGD (Abadi et al., 2016), where the gradients of the discriminator (which compares generated samples to private data) are privatized in each training step, resulting in a high overall privacy loss (Park et al., 2017; Torkzadehmahani et al., 2019; Yoon et al., 2019; Xie et al., 2018; Frigerio et al., 2019). Another challenge is that, as the gradients must have bounded norm to derive the DP guarantee, the amount of noise for privatization in DP-SGD increases proportionally to the dimension of the discriminator. Hence, these methods are typically bound to relatively small discriminators, limiting the ability to learn data distributions beyond, say, MNIST (LeCun & Cortes, 2010) or FashionMNIST (Xiao et al., 2017).

Given these challenges, the heavy machinery such as GANs and large-scale auto-encoder-based methods – capable of generating complex datasets in a non-private setting – fails to model datasets such as CIFAR-10 (Krizhevsky, 2009) or

CelebA (Liu et al., 2015) with a meaningful privacy guarantee (e.g., $\epsilon \approx 2$). Typical deep generative modeling papers have moved well beyond these datasets, but to the best of our knowledge, currently there is no DP data generation method that can produce reliable samples at a reasonable privacy level.

How can we reduce this huge gap between the performance of non-private deep generative models and that of private counterparts? We argue that we can narrow this gap by using the abundant resource of *public* data, in line with the core message of Tramer & Boneh (2021): *We simply need better features for differentially private learning*. While Tramer & Boneh demonstrated this in the context of DP classification, we aim to show the applicability of this reasoning for the more challenging problem of DP data generation, with a focus on high-dimensional image generation.

We propose to exploit public data to learn *perceptual features* (PFs) from public data, which we will use to compare synthetic and real data distributions. Following dos Santos et al. (2019), we use "perceptual features" to mean the vector of all activations of a pretrained deep network for a given data point, e.g. the hundreds of thousands of hidden activations from applying a trained deep classifier to an image. Building on dos Santos et al. (2019), who use PFs for transfer learning in natural image generation, our goal is to improve the quality of natural images generated with differential privacy constraints.

We construct a kernel on images using these powerful PFs, then train a generator by minimizing the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) between distributions (as in Harder et al., 2021; Li et al., 2015; Dziugaite et al., 2015; dos Santos et al., 2019). This scheme is non-adversarial, leading to simpler and more stable optimization; moreover, it allows us to privatize the mean embedding of the private dataset *once*, using it at each step of generator training without incurring cumulative privacy losses.

We observe in our experiments that as long as the public data contains more complex patterns than private data, e.g., transferring the knowledge learned from ImageNet as public data to generate CIFAR-10 images as private data, the learned features from public data are useful enough to generate good synthetic data. We successfully generate reasonable samples for CIFAR-10, CelebA, MNIST, and FashionMNIST in high-privacy regimes. We also theoretically analyze the effect of privatizing our loss function, helping understand the privacy-accuracy trade-offs in our method.

The main point of our paper is that features from public data are a key tool for improved DP data generation, a point we think our experiments make resoundingly; this may be "obvious", but has not been explored for image generation. Our proposed method, in particular, is a simple (which, we think, is a good thing) initial technique exploiting this idea, which outperforms simple pretraining of DP-GAN and DP-Sinkhorn (see Section 6). We hope this work will inspire future work on other ways to use public features for improving image generation with differential privacy.

2 BACKGROUND

We provide background information on maximum mean discrepancy and differential privacy.

Maximum Mean Discrepancy The MMD is a distance between distributions based on a kernel $k_\phi(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$, where ϕ maps data in $\mathcal{X}$ to a Hilbert space $\mathcal{H}$ (Gretton et al., 2012). One definition is

$$\mathrm{MMD}_{k_\phi}(P, Q) = \left\| \mathbb{E}_{x\sim P}[\phi(x)] - \mathbb{E}_{y\sim Q}[\phi(y)] \right\|_{\mathcal{H}},$$

where $\boldsymbol{\mu}_\phi(P) = \mathbb{E}_{x\sim P}[\phi(x)] \in \mathcal{H}$ is known as the (kernel) *mean embedding* of P, and is guaranteed to exist if $\mathbb{E}_{x\sim P}\sqrt{k(x, x)} < \infty$ (Smola et al., 2007). If k_ϕ is *characteristic* (Sriperumbudur et al., 2011), then $P \mapsto \boldsymbol{\mu}_\phi(P)$ is injective, so $\mathrm{MMD}_{k_\phi}(P, Q) = 0$ if and only if $P = Q$.

For a sample set $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^m \sim P^m$, the empirical mean embedding $\boldsymbol{\mu}_\phi(\mathcal{D}) = \frac{1}{m}\sum_{i=1}^m \phi(\mathbf{x}_i)$ is the "plug-in" estimator of $\boldsymbol{\mu}_\phi(P)$ using the empirical distribution of $\mathcal{D}$. Given $\tilde{\mathcal{D}} = \{\tilde{\mathbf{x}}_i\}_{i=1}^n \sim Q^n$, we can estimate $\mathrm{MMD}_{k_\phi}(P, Q)$ as the distance between empirical mean embeddings,

$$\mathrm{MMD}_{k_\phi}(\mathcal{D}, \tilde{\mathcal{D}}) = \left\| \frac{1}{m}\sum_{i=1}^m \phi(\mathbf{x}_i) - \frac{1}{n}\sum_{i=1}^n \phi(\tilde{\mathbf{x}}_i) \right\|_{\mathcal{H}}. \tag{1}$$

We would like to minimize the distance between a target data distribution P (based on samples $\mathcal{D}$) and the output distribution $Q_{g_{\boldsymbol{\theta}}}$ of a generator network $g_{\boldsymbol{\theta}}$. If the feature map is finite-dimensional and norm-bounded, following

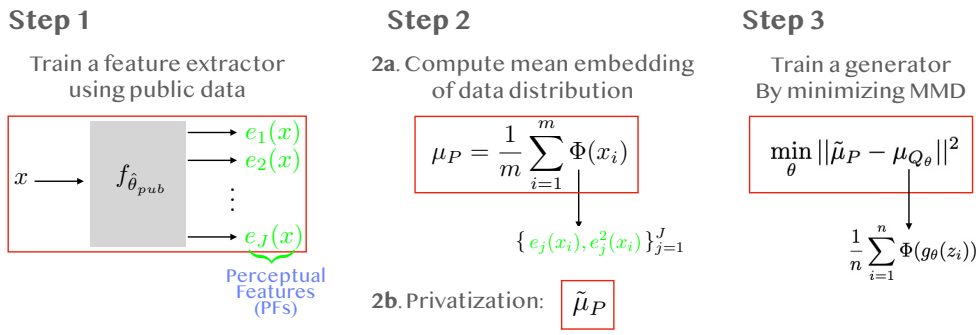

Figure 1: Three steps in *differentially private mean embedding with perceptual features (DP-MEPF)*. **Step 1:** We train a feature extractor neural network, $f_{\hat{\theta}_{pub}}$, using public data. This is a function of public data, with no privacy cost to train. A trained $f_{\hat{\theta}_{pub}}$ maps an input $\mathbf{x}$ to perceptual features (in green), the outputs of each layer. **Step 2:** We compute the mean embedding of the data distributions using a feature map consisting of the first and second moments (in green) of the perceptual features, and privatize it based on the Gaussian mechanism (see text). **Step 3:** We train a generator g_θ, which produces synthetic data from latent codes $z_i \sim \mathcal{N}(0, I)$, by minimizing the privatized MMD.

Harder et al. (2021); Vinaroz et al. (2022), we can privatize the mean embedding of the data distribution $\boldsymbol{\mu}_\phi(\mathcal{D})$ with a known DP mechanism such as the Gaussian or Laplace mechanisms, to be discussed shortly. As the summary of the real data does not change over the course of a generator training, we only need to privatize $\boldsymbol{\mu}_\phi(\mathcal{D})$ once.

Differential privacy A mechanism $\mathcal{M}$ is (ϵ, δ)-DP for a given $\epsilon \geq 0$ and $\delta \geq 0$ if and only if

$$\Pr[\mathcal{M}(\mathcal{D}) \in S] \leq e^\epsilon \cdot \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta$$

for all possible sets of the mechanism's outputs S and all neighbouring datasets $\mathcal{D}$, $\mathcal{D}'$ that differ by a single entry. One of the most well-known and widely used DP mechanisms is the *Gaussian mechanism*. The Gaussian mechanism adds a calibrated level of noise to a function $\mu : \mathcal{D} \mapsto \mathbb{R}^p$ to ensure that the output of the mechanism is (ϵ, δ)-DP: $\widetilde{\mu}(\mathcal{D}) = \mu(\mathcal{D}) + n$, where $n \sim \mathcal{N}(0, \sigma^2\Delta_\mu^2\mathbf{I}_p)$. Here, σ is often called a privacy parameter, which is a function[1] of ϵ and δ. Δ_μ is often called the *global sensitivity* (dwo), which is the maximum difference in L_2-norm given two neighbouring $\mathcal{D}$ and $\mathcal{D}'$, $||\mu(\mathcal{D}) - \mu(\mathcal{D}')||_2$. In this paper, we will use the Gaussian mechanism to ensure the mean embedding of the data distribution is DP.

3 METHOD

In this paper, to transfer knowledge from public to private data distributions, we construct a particular kernel k_Φ to use in Equation 1 based on *perceptual features* (PFs).

3.1 MMD with perceptual features as a feature map

We call our proposed method *Differentially Private Mean Embeddings with Perceptual Features (DP-ME**P**F)*, analogous to the related method DP-ME**R**F (Harder et al., 2021). We use high-dimensional, over-complete perceptual features from a feature extractor network pre-trained on a public dataset, as illustrated in **Step 1** of Figure 1. Given a vector input $\mathbf{x}$, the pre-trained feature extractor network outputs the perceptual features from each layer, where the jth layer's PF is denoted by $\mathbf{e}_j(\mathbf{x})$. Each of the J layers' perceptual features is of a different length, $\mathbf{e}_j(\mathbf{x}) \in \mathbb{R}^{d_j}$; the total dimension of the perceptual feature vector is $D = \sum_{j=1}^{J} d_j$.

As illustrated in **Step 2** in Figure 1, we use those PFs to form our feature map $\Phi(\mathbf{x}) := [\boldsymbol{\phi}_1(\mathbf{x}), \boldsymbol{\phi}_2(\mathbf{x})]$, where the first part comes from a concatenation of PFs from all the layers: $\boldsymbol{\phi}_1(\mathbf{x}) = [\mathbf{e}_1(\mathbf{x}), \cdots, \mathbf{e}_J(\mathbf{x})]$, while the second part comes from their squared values: $\boldsymbol{\phi}_2(\mathbf{x}) = [\mathbf{e}_1^2(\mathbf{x}), \cdots, \mathbf{e}_J^2(\mathbf{x})]$, where $\mathbf{e}_j^2(\mathbf{x})$ means each entry of $\mathbf{e}_j(\mathbf{x})$ is squared. Using

[1]The relationship can be numerically computed by packages like `auto-dp` (Wang et al., 2019), among other methods.

this feature map, we then construct the mean embedding of a data distribution given the data samples $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^m$:

$$\boldsymbol{\mu}_P(\mathcal{D}) = \begin{bmatrix} \boldsymbol{\mu}_P^{\phi_1}(\mathcal{D}) \\ \boldsymbol{\mu}_P^{\phi_2}(\mathcal{D}) \end{bmatrix} = \begin{bmatrix} \frac{1}{m}\sum_{i=1}^m \boldsymbol{\phi}_1(\mathbf{x}_i) \\ \frac{1}{m}\sum_{i=1}^m \boldsymbol{\phi}_2(\mathbf{x}_i) \end{bmatrix}. \tag{2}$$

Lastly (**Step 3** in Figure 1), we will train a generator $g_{\boldsymbol{\theta}}$ that maps latent vectors $\mathbf{z}_i \sim \mathcal{N}(0, I)$ to a synthetic data sample $\tilde{\mathbf{x}}_i = g_{\boldsymbol{\theta}}(\mathbf{z}_i)$; we need to find good parameters $\boldsymbol{\theta}$ for the generator. In non-private settings, we estimate the generator's parameters by minimizing an estimate of $\mathrm{MMD}^2_{k_\Phi}(P, Q_{g_{\boldsymbol{\theta}}})$, using $\tilde{\mathcal{D}} = \{\tilde{\mathbf{x}}_i\}$ in Equation 1, similar to Dziugaite et al. (2015); Li et al. (2015); dos Santos et al. (2019). In private settings, we privatize $\mathcal{D}$'s mean embedding to $\tilde{\boldsymbol{\mu}}_\phi(\mathcal{D})$ with the Gaussian mechanism (details below), and minimize

$$\widetilde{\mathrm{MMD}}^2_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}) = \left\|\tilde{\boldsymbol{\mu}}_\phi(\mathcal{D}) - \boldsymbol{\mu}_\phi(\tilde{\mathcal{D}})\right\|^2. \tag{3}$$

Extending our framework to generate both labels and input images is straightforward. As done by Harder et al. (2021), we construct a mean embedding for each class-conditional input distribution, such that the size of the final mean embedding is D (number of perceptual features) by the number of classes if we use only the first moment, or $2D$ by the number of classes if we use the first two moments. This is exactly the conditional mean embedding with a discrete kernel on the class label (Song et al., 2013).

A natural question that arises is whether the MMD using the PFs is a metric: if $\mathrm{MMD}_{k_\Phi}(P, Q) = 0$ only if $P = Q$. As PFs have a finite-dimensional embedding, we in fact know this cannot be the case (Sriperumbudur et al., 2011). Thus, there exists *some* pair of distributions which our MMD cannot distinguish. However, given that linear functions in perceptual feature spaces can obtain excellent performance on nearly any natural image task (as observed in transfer learning), it seems that PFs are "nearly" universal for natural distributions of images (dos Santos et al., 2019). Thus we expect the MMD with this kernel to do a good job of distinguishing "natural" distributions from one another, though the possibility of "adversarial attacks" perhaps remains.

A more important question in our context is whether this MMD serves as a good loss for training a generator, and whether the resulting synthetic data samples are reasonably faithful to the original data samples. Our experiments in Section 6, as well as earlier work by dos Santos et al. (2019) in non-private settings, imply that it is.

Privatization of mean embedding We privatize the mean embedding of the data distribution only once, and reuse it repeatedly during the training of the generator $g_{\boldsymbol{\theta}}$. We use the Gaussian mechanism to separately privatize the first and second parts of the feature map. We normalize each type of perceptual features such that $\|\boldsymbol{\phi}_1(\mathbf{x}_i)\|_2 = 1$ and $\|\boldsymbol{\phi}_2(\mathbf{x}_i)\|_2 = 1$ for each sample $\mathbf{x}_i$. After this change, the sensitivity of each part of the mean embedding is

$$\max_{\mathcal{D}, \mathcal{D}' \text{ s.t. } |\mathcal{D} - \mathcal{D}'| = 1} \|\boldsymbol{\mu}_{\boldsymbol{\phi}_t}(\mathcal{D}) - \boldsymbol{\mu}_{\boldsymbol{\phi}_t}(\mathcal{D}')\|_2 \leq \frac{2}{m}, \tag{4}$$

where $\boldsymbol{\mu}_{\boldsymbol{\phi}_t}(\mathcal{D})$ denotes the two parts of the mean embedding for $t = 1, 2$. Using these sensitivities, we add Gaussian noise to each part of the mean embedding, obtaining

$$\tilde{\boldsymbol{\mu}}_\Phi(\mathcal{D}) = \begin{bmatrix} \tilde{\boldsymbol{\mu}}_{\boldsymbol{\phi}_1}(\mathcal{D}) \\ \tilde{\boldsymbol{\mu}}_{\boldsymbol{\phi}_2}(\mathcal{D}) \end{bmatrix} = \begin{bmatrix} \frac{1}{m}\sum_{i=1}^m \boldsymbol{\phi}_1(\mathbf{x}_i) + \mathbf{n}_1 \\ \frac{1}{m}\sum_{i=1}^m \boldsymbol{\phi}_2(\mathbf{x}_i) + \mathbf{n}_2 \end{bmatrix}, \tag{5}$$

where $\mathbf{n}_t \sim \mathcal{N}(0, \frac{4\sigma^2}{m^2} I)$ for $t = 1, 2$.

Since we are using the Gaussian mechanism twice, we simply compose the privacy losses from each mechanism. More precisely, given a desired privacy level ϵ, δ, we use the package of Wang et al. (2019) to find the corresponding σ for the two Gaussian mechanisms.

3.2 Differentially private early stopping

On some datasets (CelebA and Cifar10) we observe that the generated sample quality deteriorates if the model is trained for too many iterations in high-privacy settings. This is indicated by a steady increase in FID score (Heusel et al., 2017),

and likely due to overfitting to the static noisy embedding. Since the FID score is based on the training data, simply choosing the iteration with the best FID score after training has completed would violate privacy.

Privatizing the FID score requires privatizing the covariance of the output of the final pooling layer in the Inception network, which is quite sensitive. Instead, we privatize the first and second moment of data embeddings as in Equation 2, but using only the output of the final pooling layer in the Inception network. We then use this quantity as a private proxy for FID, and select the iteration with the lowest score. To minimize the privacy cost, we choose a larger noise parameter than for the main objective: $\sigma_{stopping} = 10\sigma$, where σ is the noise scale for privatizing each part of the data mean embeddings, works well. Again, we compose these σs with the analysis of Wang et al. (2019).

4 THEORETICAL ANALYSIS

We now bound the effect of adding noise to our loss function, showing that asymptotically our noise does not hurt the rate at which our model converges to the optimal model.

Section A proves full finite-sample versions of all of the following bounds, which are stated here using $\mathcal{O}_P$ notation for simplicity. $X = \mathcal{O}_P(A_n)$ essentially means that X is $\mathcal{O}(A_n)$ with probability at least $1 - \rho$ for any constant choice of failure probability $\rho > 0$.

The full version in the supplementary material is also ambivalent to the choice of covariance for the noise variable $\mathbf{n}$, allowing in particular analysis of DP-MEPF based either on one or two moments of PFs. (The full version gives a slightly more refined treatment of the two-moment case, but the difference is typically not asymptotically relevant.)

To begin, we use standard concentration results on Gaussians to establish that the privatized MMD is close to the non-private MMD:

Proposition 4.1. *Given datasets $\mathcal{D}$ and $\tilde{\mathcal{D}}$, the absolute difference between the privatized and non-private squared MMDs, a random function of only $\mathbf{n}$, satisfies* $\left|\widetilde{\mathrm{MMD}}^2_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}) - \mathrm{MMD}^2_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}})\right| = \mathcal{O}_P\left(\frac{\sigma^2}{m^2}D + \frac{\sigma}{m}\mathrm{MMD}_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}})\right)$.

One key quantity in the bound is σ/m, the ratio of the noise scale σ (inversely proportional to ε) to the number of observed (private) data points m. Note that σ depends only on the given privacy level, not on m, so the error becomes zero as long as $m \to \infty$. In the second term, σ/m is multiplied by the (non-private, non-squared) MMD, which is bounded for our features, but for good generators (where our optimization hopefully spends most of its time) this term will also be nearly zero. The other term accounts for adding independent noise to each of the D feature dimensions; although D is typically large, so is m^2. Having $m = 50\text{K}$ private samples, e.g. for CIFAR-10, allows for a strong error bound as long as $D\sigma^2 \ll 625\text{M}$.

The above result is for a fixed pair of datasets. Because we only add noise $\mathbf{n}$ once, across all possible comparisons, we can use this to obtain a bound uniform over all possible generator distributions, in particular implying that the minimizer of the privatized MMD approximately minimizes the original, non-private MMD:

Proposition 4.2. *Fix a target dataset $\mathcal{D}$. For each $\boldsymbol{\theta}$ in some set Θ, fix a corresponding $\tilde{\mathcal{D}}_{\boldsymbol{\theta}}$; in particular, $\Theta = \mathbb{R}^p$ could be the set of all generator parameters, and $\tilde{\mathcal{D}}_{\boldsymbol{\theta}}$ either the outcome of running a generator $g_{\boldsymbol{\theta}}$ on a fixed set of "seeds," $\tilde{\mathcal{D}}_{\boldsymbol{\theta}} = \{g_{\boldsymbol{\theta}}(\mathbf{z}_i)\}_{i=1}^n$, or the full output distribution of the generator $Q_{g_{\boldsymbol{\theta}}}$. Let $\widetilde{\boldsymbol{\theta}} \in \arg\min_{\theta\in\Theta} \widetilde{\mathrm{MMD}}^2_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}_\theta)$ be the private minimizer, and $\widehat{\boldsymbol{\theta}} \in \arg\min_{\theta\in\Theta} \mathrm{MMD}^2_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}_\theta)$ the non-private minimizer. Then* $\mathrm{MMD}^2_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}_{\widetilde{\boldsymbol{\theta}}}) - \mathrm{MMD}^2_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}_{\widehat{\boldsymbol{\theta}}}) = \mathcal{O}_P\left(\frac{\sigma^2 D}{m^2} + \frac{\sigma\sqrt{D}}{m}\right)$.

The second term of this bound will generally dominate; it arises from uniformly bounding the $\frac{\sigma}{m}\mathrm{MMD}_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}_{\boldsymbol{\theta}})$ term of Proposition 4.1 over all possible $\tilde{\mathcal{D}}_{\boldsymbol{\theta}}$. This approach, although the default way to prove this type of bound, misses that $\mathrm{MMD}_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}_{\boldsymbol{\theta}})$ is hopefully small for $\widetilde{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{\theta}}$. We can in fact take advantage of this to provide an "optimistic" rate (Srebro et al., 2010; Zhou et al., 2021) that achieves faster convergence if the generator is capable of matching the target features (an "interpolating" regime):

Proposition 4.3. *In the setting of Proposition 4.2,* $\mathrm{MMD}^2_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}_{\widetilde{\boldsymbol{\theta}}}) - \mathrm{MMD}^2_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}_{\widehat{\boldsymbol{\theta}}}) = \mathcal{O}_P\left(\frac{\sigma^2 D}{m^2} + \frac{\sigma\sqrt{D}}{m}\mathrm{MMD}_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}}_{\widehat{\boldsymbol{\theta}}})\right)$. Note that this bound implies the previous one, since $\mathrm{MMD}_{k_\Phi}(\mathcal{D}, \tilde{\mathcal{D}})$ is

bounded. But in the case where the generator is capable of exactly matching the features of the target distribution, the second term becomes zero, and the rate with respect to m is greatly improved.

In either regime, our approximate minimization of the empirical MMD is far faster than the rate at which minimizing the empirical $\mathrm{MMD}(\mathcal{D}, Q_{g_\theta})$ converges to minimizing the true, distribution-level $\mathrm{MMD}(P, Q_{g_\theta})$: the known results there (e.g. Dziugaite et al., 2015, Theorem 1) give a $1/\sqrt{m}$ rate, compared to our $1/m$ or even $1/m^2$.

We show that minimizing DP-MEPF's loss actually pays *no* asymptotic penalty for privacy (especially when a perfect generator exists), with the privacy loss dwarfed by the statistical error for large datasets; this essentially agrees with experiments (see Section 6). This is not the case for all DP methods, and other DP generation papers didn't prove any such guarantees: DP-Sinkhorn only proved privacy, and DP-MERF showed only a much weaker guarantee (its gradient is asymptotically unbiased).

5 RELATED WORK

Initial work on differentially private data generation assumed strong constraints on the type of data and the intended use of the released data (Snoke & Slavković, 2018; Mohammed et al., 2011; Xiao et al., 2010; Hardt et al., 2012; Zhu et al., 2017). While these studies provide theoretical guarantees on the utility of the synthetic data, they typically do not scale to our goal of large-scale image data generation.

Recently, several papers focused on discrete data generation with limited domain size (Zhang et al., 2017; Qardaji et al., 2014; Chen et al., 2015; Zhang et al., 2021). These methods learn the correlation structure of small subsets of features and privatize them in order to produce differentially private synthetic data samples. These methods often require discretization of the data and have limited scalability, so are also unsuitable for high-dimensional image data generation.

More recently, however, a new line of work has emerged that adopt the core ideas from the recent advances in deep generative models for a broad applicability of synthetic data with differential privacy constraints. The majority of this work (Xie et al., 2018; Torkzadehmahani et al., 2019; Frigerio et al., 2019; Yoon et al., 2019; Chen et al., 2020) uses generative adversarial networks (GANs; Goodfellow et al., 2014) along with some form of DP-SGD (Abadi et al., 2016). Other works in this line include PATE-GAN based on the private aggregation of teacher ensembles (Papernot et al., 2017) and variational autoencoders (Acs et al., 2018).

The closest prior work to the proposed method is DP-MERF (Harder et al., 2021), where the kernel mean embeddings are constructed using random Fourier features (Rahimi & Recht, 2008). A recent variant of DP-MERF uses Hermite polynomial-based mean embeddings (Vinaroz et al., 2022). Unlike these methods, we use the perceptual features from a pre-trained network to construct kernel mean embeddings. Neither previous method applies to the perceptual kernels used here, so their empirical results are far worse (as we'll see shortly). Our theoretical analysis is also much more extensive: they only proved a bound on the expected error between the private and non-private empirical MMD for a fixed pair of datasets.

More recently, a similar work to DP-MERF utilizes the Sinkhorn divergence for private data generation (Cao et al., 2021), which performs similarly to DP-MERF when the cost function is the L2 distance with a large regularizer. Another related work proposes to use the characteristic function and an adversarial re-weighting objective (Liew et al., 2022) in order to improve the generalization capability of DP-MERF.

A majority of these related methods were evaluated only on relatively simple datasets such as MNIST and FashionMNIST. Even so, the DP-GAN-based methods mostly require a large privacy budget of $\epsilon \approx 10$ to generate synthetic data samples that are reasonably close to the real data samples. Our method goes far beyond this quality with much more stringent privacy constraints, as we will now see.

6 EXPERIMENTS

We will now compare our method to state-of-the-art methods for DP data generation.

Table 1: Downstream accuracies by Logistic regression and MLP, evaluated on the generated data samples using MNIST and FashionMNIST as private data and SVHN and CIFAR-10 as public data, respectively. In all cases, we set $\epsilon = 10, \delta = 10^{-5}$. In our method, we used both features ϕ_1, ϕ_2.

		DP-MEPF	DP-Sinkhorn (Cao et al., 2021)	GS-WGAN (Chen et al., 2020)	DP-MERF (Harder et al., 2021)	DP-HP (Vinaroz et al., 2022)
MNIST	LogReg	**83**	83	79	79	81
	MLP	**90**	83	79	78	82
F-MNIST	LogReg	**76**	75	68	76	73
	MLP	**76**	75	65	75	71

Datasets. We considered four image datasets[2] of varying complexity. We started with the commonly used datasets MNIST (LeCun & Cortes, 2010) and FashionMNIST (Xiao et al., 2017), where each consist of 60,000 28×28 pixel grayscale images depicting hand-written digits and items of clothing, respectively, sorted into 10 classes. We also looked at the more complex CelebA (Liu et al., 2015) dataset, containing 202,599 color images of faces which we scale to sizes of 32×32 or 64×64 pixels and treat as unlabeled. We also study CIFAR-10 (Krizhevsky, 2009), a 50,000-sample dataset containing 32×32 color images of 10 classes of objects, including vehicles like ships and trucks, and animals such as horses and birds.

Implementation. We implemented our code for all the experiments in PyTorch (Paszke et al., 2019), using the `auto-dp` package[3] (Wang et al., 2019) for the privacy analysis. Following Harder et al. (2021), we used the generator that consists of two fully connected layers followed by two convolutional layers with bilinear upsampling, for generating both MNIST and FashionMNIST datasets. For MNIST, we used the SVHN dataset as public data to pre-train ResNet18 (He et al., 2016), from which we took the perceptual features. For FashionMNIST, we used perceptual features from a ResNet18 trained on CIFAR-10. For CelebA and CIFAR-10, we followed dos Santos et al. (2019) in using perceptual features from a pre-trained VGG (Simonyan & Zisserman, 2014) on ImageNet, and a ResNet18-based generator. Further implementation details are given in the supplementary material, which also studies how different public datasets and feature extractors impact the performance.

Evaluation metric. Evaluating the quality of generated data is a challenging problem of its own. We use two conventional measures. The first is the *Frechet Inception Distance (FID)* score (Heusel et al., 2017), which directly measures the quality of the generated samples. The FID score correlates with human evaluations of visual similarity to the real data, and is commonly used in deep generative modelling. We computed FID scores with the `pytorch_fid` package (Seitzer, 2020), based on $5\,000$ generated samples, matching dos Santos et al. (2019). As discussed in Section 3.2, we use a private proxy for FID for early stopping, while the FID scores we report in this section are non-DP measures of our final model for fair comparison to other existing methods. The second metric we use is the accuracy of downstream classifiers, trained on generated datasets and then test on the real data test sets (used by Chen et al., 2020; Torkzadehmahani et al., 2019; Yoon et al., 2019; Chen et al., 2020; Harder et al., 2021; Cao et al., 2021). This test accuracy indicates how well the downstream classifiers generalize from the synthetic to the real data distribution and thus, the utility of using synthetic data samples instead of the real ones. We computed the downstream accuracy on MNIST and FashionMNIST using the logistic regression and MLP classifiers from scikit-learn (Pedregosa et al., 2011). For CIFAR-10, we used ResNet9 taken from FFCV[4] (Leclerc et al., 2022).

In all experiments, we tested non-private training and settings with various levels of privacy, ranging from $\epsilon = 10$ (no meaningful guarantee) to $\epsilon = 0.2$ (strong privacy guarantee). We always set $\delta = 10^{-5}$. In DP-MEPF, we also tested cases based on embeddings with only the first moment, written (ϕ_1), and using the first two moments, written (ϕ_1, ϕ_2). Each value in all tables is an average of 3 or more runs; standard deviations are in the supplementary material.

Since we are unaware of any prior work on DP data generation for image data using auxiliary datasets, we instead mostly compare to recent methods which do not access auxiliary data. As expected, due to the advantage of non-private data our approach outperforms these methods by a significant margin on the more complex datasets. As a simple

[2]Dataset licenses: MNIST: CC BY-SA 3.0; FashionMNIST:MIT; CelebA: see `https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`; Cifar10: MIT

[3]`https://github.com/yuxiangw/autodp`

[4]`https://github.com/libffcv/ffcv/blob/main/examples/cifar/train_cifar.py`

Table 2: Downstream accuracies of our method for MNIST and FashionMNIST at varying values of ϵ.

		MNIST				**FashionMNIST**			
		$\epsilon = 5$	$\epsilon = 2$	$\epsilon = 1$	$\epsilon = 0.2$	$\epsilon = 5$	$\epsilon = 2$	$\epsilon = 1$	$\epsilon = 0.2$
MLP	DP-MEPF (ϕ_1, ϕ_2)	90	89	89	80	76	75	75	70
	DP-MEPF (ϕ_1)	88	88	87	77	75	76	75	69
LogReg	DP-MEPF (ϕ_1, ϕ_2)	83	83	82	76	75	76	75	73
	DP-MEPF (ϕ_1)	81	80	79	72	75	76	76	72

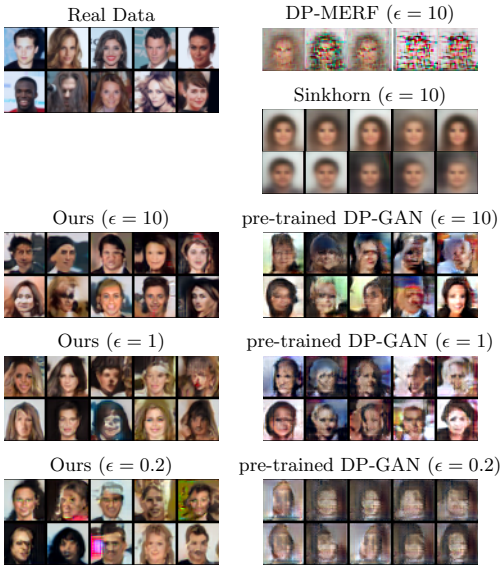

Figure 2: Synthetic 32×32 CelebA samples generated at different levels of privacy. Samples for DP-MERF and DP-Sinkhorn are taken from Cao et al. (2021). The pre-trained GAN is our baseline utilizing public data. Even at $\epsilon = 0.2$, DP-MEPF (ϕ_1, ϕ_2) yields samples of higher visual quality than the comparison methods.

baseline based on public data, we also pretrain a GAN on a downscaled version of ImageNet, at 32×32, and fine-tune this model with DP-SGD on CelebA and Cifar10. We use architectures based on ResNet9 with group normalization (Wu & He, 2018) for both generator and discriminator. As suggested by Anonymous (2023), we update the generator at a lower frequency than the discriminator and use increased minibatch sizes. Further details can be found in the supplementary material.

Figure 3: Synthetic 64×64 CelebA samples generated at different levels of privacy with DP-MEPF (ϕ_1, ϕ_2).

MNIST and FashionMNIST. We compare DP-MEPF to existing methods on the most common settings used in the literature, MNIST and FashionMNIST at $\epsilon = 10$, in Table 1. For an MLP on MNIST, DP-MEPF's samples far outperform other methods for logistic regression and both classifiers on FashionMNIST, scores match or slightly exceed those of existing models. This might be because the domain shift between public dataset (CIFAR-10, color images of scenes) and private dataset (FashionMNIST, grayscale images of fashion items) is too large, or because the task is simple enough that random features as found in DP-MERF or DP-HP are already good enough. This will change as we proceed to more complex datasets.

Table 3: CelebA FID scores (lower is better) for images of resolution 32×32 and 64×64.

		$\epsilon = 10$	$\epsilon = 5$	$\epsilon = 2$	$\epsilon = 1$	$\epsilon = 0.5$	$\epsilon = 0.2$
32	DP-MEPF $(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$	15.1	14.3	13.9	14.9	14.4	19.3
	DP-MEPF $(\boldsymbol{\phi}_1)$	11.7	12.1	12.6	13.2	14.4	18.1
	DP-GAN (pre-trained)	40.3	49.8	57.5	53.3	72.8	148.2
64	DP-MEPF $(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$	13.0	13.1	13.2	13.5	15.5	24.8
	DP-MEPF $(\boldsymbol{\phi}_1)$	11.7	11.7	11.6	13.0	16.2	27.3

Table 4: Two examples of beneficial early stopping: For CelebA at 64×64 resolution and labeled Cifar10, DP-MEPF $(\boldsymbol{\phi}_1)$ sample quality (measured in FID) degrades with long training in high privacy settings (here $\epsilon \leq 1$). This makes the final model at the end of training a poor choice. Our DP selection of the best iteration via proxy stays close to the optimal choice.

		$\epsilon = 1$	$\epsilon = 0.5$	$\epsilon = 0.2$
CelebA 64×64	Best FID (not DP)	12.3	15.7	25.5
	DP proxy for FID	13.0	16.2	27.3
	At the end of training	12.6	17.6	97.3
Cifar10 (labeled)	Best FID (not DP)	38.0	78.4	350.3
	DP proxy for FID	39.0	78.4	469.3
	At the end of training	311.8	354.4	371.7

Table 2 shows that downstream test accuracy only starts to drop in high privacy regimes, $\epsilon < 1$, due to the low sensitivity of $\boldsymbol{\mu}_\phi$. Samples for visual comparison between methods are included in the supplementary material.

CelebA Figure 2 shows that previous attempts to generate CelebA samples without auxiliary data using DP-MERF or DP-Sinkhorn have only managed to capture very basic features of the data. Each sample depicts a face, but offers no details or variety. DP-MEPF produces more accurate samples at the same 32×32 resolution, which is also reflected in improved FID scores of around 12, while DP-Sinkhorn achieves and FID of 190. As the dataset has over $200\,000$ samples, the feature embeddings have low sensitivity, and offer similar quality between $\epsilon = 10$ and $\epsilon = 1$, although quality begins to drop at $\epsilon < 1$. Samples for 64×64 images are shown in Figure 3, with similar quality; Table 3 gives FID scores for both resolutions at varying ϵ.

We acknowledge that $\delta = 10^{-5}$ is a bad choice for CelebA, as at $n = 202,599$ samples, $\delta' = 1/203{,}000$ is appropriate. The main reason we stick to $\delta = 10^{-5}$ in these experiments is that other existing methods were all tested at this level of privacy guarantee. However, this change in δ will barely affect the DP guarantee. For (ϕ_1, ϕ_2), (10, δ)-DP also implies (10.21, δ')-DP and (0.2, δ)-DP implies (0.21, δ')-DP. Guaranteeing (10, δ')-DP and (0.2, δ')-DP requires a respective increase in noise factor σ of 2.1% and 5%, which will have little effect.

Because DP-Sinkhorn is the best-performing method without public data, we perform experiments on DP-Sinkhorn, pretraining it non-DP on ImageNet32 and fine-tuning with DP on CelebA ($\epsilon = 10$). After seeing no improvement, we tested non-DP fine-tuning and still saw no improvements beyond what is shown in Figure 4; we tried both BigGan- and ResNet18-based generators with hyperparameter grid searches. DP-Sinkhorn only compares features at image-level, without domain-specific priors, and it appears that even non-DP the method is not powerful enough to model image data beyond MNIST. (A DP-MEPF analogue that extracts features learned from public data might help, but this would be a novel method beyond scope for comparison.) DP-MERF is similarly limited by its random features, not DP noise, as shown by non-DP versions matching $\epsilon = 10$ performance.

Differentially private early stopping. For CelebA and Cifar10, we use DP early stopping as explained in Section 3.2 with a privacy parameter ten times larger than the σ used for the training objective. Keeping (ϵ, δ) fixed, this additional release results only in a small increase in σ, and gives us a simple way for choosing the best iteration. In Table 4, we compare the true best FID, the FID picked by our private proxy, and the FID at the end of training to illustrate the advantage in high DP settings. FID scores were computed every $5\,000$ iterations, while the model trained for $200\,000$ iterations in total.

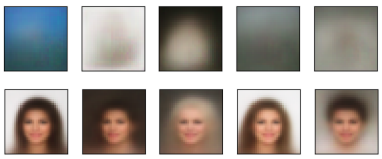

Figure 4: Samples from non-DP Sinkhorn. Top: ImageNet32. Bottom: CelebA after pretraining.

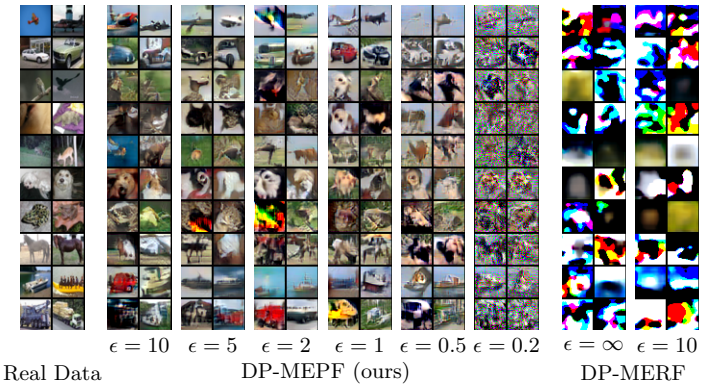

Figure 5: Labeled samples from DP-MEPF (ϕ_1, ϕ_2) and DP-MERF (Harder et al., 2021).

Figure 6: CIFAR-10 samples. Image quality degrades less with high privacy guarantees for unlabelled generation.

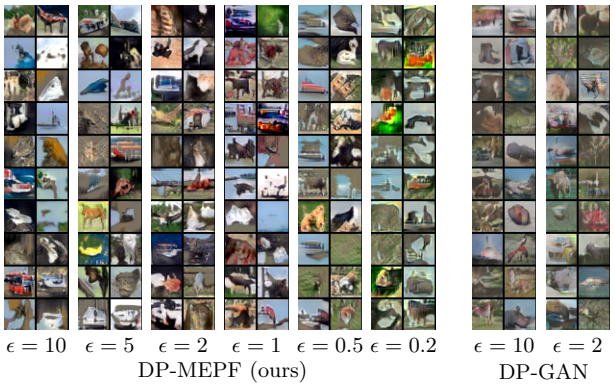

Figure 7: Unlabeled CIFAR-10 samples from DP-MEPF (ϕ_1, ϕ_2) and DP-GAN.

CIFAR-10 Finally, we investigate a dataset which has not been covered in DP data generation. While CelebA depicts a centered face in every image, CIFAR-10 includes 10 visually distinct object classes, which raises the required minimum quality of samples to somewhat resemble the dataset. At only 5 000 samples per class, the dataset is also significantly smaller, which poses a challenge in the private setting.

Figure 5 shows that DP-MEPF is capable of producing labelled private data (generating both labels and input images together) resembling the real data, but the quality does suffer in high privacy settings. This is also reflected in the FID scores (Table 5): at $\epsilon \leq 1$ labeled DP-MEPF scores deteriorate at a much quicker rate than the unlabeled counterpart. As the unlabeled embedding dimension is smaller by a factor of 10 (the number of classes), it is easier to release privately and retains some semblance of the data even in the highest privacy settings, as shown in Figure 7. The FID scores of our pre-trained DP-GAN baseline consistently exceed our results by 10 or more points. These scores are better than the DP-GAN results for CelebA, likely because 32×32 ImageNet is very similar to Cifar10. Nonetheless, the high privacy cost of DP-SGD makes DP-GAN a poor fit for a dataset of this complexity and limited size.

Table 5: FID scores for synthetic CIFAR-10 data; labeled generates both labels and images.

		$\epsilon = 10$	$\epsilon = 5$	$\epsilon = 2$	$\epsilon = 1$	$\epsilon = 0.5$	$\epsilon = 0.2$
unlabeled	DP-MEPF (ϕ_1, ϕ_2)	27.1	24.9	26.0	27.2	34.8	56.6
	DP-MEPF (ϕ_1)	26.8	25.9	28.9	32.0	38.6	53.9
	DP-GAN	37.8	39.0	40.6	54.1	60.3	63.0
labeled	DP-MEPF (ϕ_1, ϕ_2)	26.6	27.6	27.6	38.6	64.4	325.0
	DP-MEPF (ϕ_1)	27.1	27.7	28.7	39.0	78.4	469.3

In Table 6 we show the test accuracy of models trained synthetic datasets applied to real data. While there is still a large gap between the 88.3% accuracy on the real data and our results, DP-MEPF achieves nontrivial results around 50% for $\epsilon = 10$, which slowly degrade as privacy is increased. While the drop in sample quality due to high privacy is quite substantial, it is less of a problem in the unlabelled case, since our embedding dimension is smaller by a factor of 10 (the number of classes) and thus easier to release privately.

Table 6: Test accuracies (higher is better) of ResNet9 trained on CIFAR-10 synthetic data with varying privacy guarantees. When trained on real data, test accuracy is 88.3%

	$\epsilon = 10$	$\epsilon = 5$	$\epsilon = 2$	$\epsilon = 1$	$\epsilon = 0.5$	$\epsilon = 0.2$
DP-MEPF (ϕ_1)	51.0	48.5	42.5	29.4	19.4	13.8
DP-MEPF (ϕ_1, ϕ_2)	48.9	47.9	38.7	28.9	19.7	12.4
DP-MERF	13.2	13.4	13.5	13.8	13.1	10.4

7 DISCUSSION

We have demonstrated the advantage of using auxiliary public data in DP data generation. Our method DP-MEPF takes advantage of features from pre-trained classifiers that are readily available, and allows us to tackle datasets like CelebA and CIFAR-10, which have been unreachable for private data generation up to this point.

There are several avenues to extend our method in future work, in particular finding better options for the encoder features: the choice of VGG19 by dos Santos et al. (2019) works well in private settings, but a lower-dimensional embedding that still works well for training generative models – perhaps based on some kind of pruning scheme – might help reduce the sensitivity of $\boldsymbol{\mu}_\phi$ and improve quality.

Training other generative models such as GANs or VAEs with pretrained components is also exploring further than our initial attempt here. It may also be possible to take a "middle ground" and introduce some adaptation for features in DP-MEPF, to allow for more powerful, GAN-like models, without suffering too much privacy loss. In the non-private generative modelling community, this has proved important, but the challenge will be to do so while limiting the number of DP releases to allow modelling with, e.g., $\epsilon \leq 2$.

References

Martin Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 308–318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318.

Gergely Acs, Luca Melis, Claude Castelluccia, and Emiliano De Cristofaro. Differentially private mixture of generative neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 31(6):1109–1121, 2018.

Anonymous. Private GANs, revisited. In *Submitted to The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=QEmn_Hvh7j8`. Under review.

Differential Privacy Team Apple. Learning with privacy at scale, 2017. URL `https://machinelearning.apple.com/research/learning-with-privacy-at-scale`.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence. In *Neural Information Processing Systems (NeurIPS)*, 2021.

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

Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. Differentially private high-dimensional data publication via sampling-based inference. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 129–138, 2015.

Cícero Nogueira dos Santos, Youssef Mroueh, Inkit Padhi, and Pierre L. Dognin. Learning implicit generative models by matching perceptual features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4460–4469. IEEE, 2019. doi: 10.1109/ICCV.2019.00456.

Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via Maximum Mean Discrepancy optimization. In *UAI*, 2015.

Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In *ICT Systems Security and Privacy Protection - 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings*, pp. 151–164, 2019. doi: 10.1007/978-3-030-22312-0_11.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

Frederik Harder, Kamil Adamczewski, and Mijung Park. DP-MERF: Differentially private mean embeddings with random features for practical privacy-preserving data generation. In *AISTATS*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1819–1827. PMLR, 2021.

Moritz Hardt, Katrina Ligett, and Frank Mcsherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems 25*, pp. 2339–2347. Curran Associates, Inc., 2012.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pp. 1302–1338, 2000.

Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Madry. ffcv. `https://github.com/libffcv/ffcv/`, 2022.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Yujia Li, Kevin Swersky, and Richard S. Zemel. Generative moment matching networks. In *ICML*, 2015.

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

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.

Noman Mohammed, Rui Chen, Benjamin C.M. Fung, and Philip S. Yu. Differentially private data release for data mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pp. 493–501, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0813-7. doi: 10.1145/2020408.2020487.

National Conference of State Legislatures. Differential privacy for census data, 2021. URL https://www.ncsl.org/research/redistricting/differential-privacy-for-census-data-explained.aspx.

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

Mijung Park, James Foulds, Kamalika Choudhary, and Max Welling. DP-EM: Differentially Private Expectation Maximization. In *AISTATS*, volume 54 of *Proceedings of Machine Learning Research*, pp. 896–904, Fort Lauderdale, FL, USA, April 2017. PMLR.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Wahbeh Qardaji, Weining Yang, and Ninghui Li. Priview: practical differentially private release of marginal contingency tables. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 1435–1446, 2014.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pp. 1177–1184, 2008.

Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.2.1.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

A. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *ALT*, pp. 13–31, 2007.

Joshua Snoke and Aleksandra Slavković. pmse mechanism: differentially private synthetic data with maximal distributional similarity. In *International Conference on Privacy in Statistical Databases*, pp. 138–159. Springer, 2018.

Le Song, Kenji Fukumizu, and Arthur Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.

Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Optimistic rates for learning with a smooth loss, 2010.

Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12(7), 2011.

Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.

Margarita Vinaroz, Mohammad-Amin Charusaie, Frederik Harder, Kamil Adamczewski, and Mi Jung Park. Hermite polynomial features for private data generation. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22300–22324. PMLR, 2022.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *AISTATS*, 2019.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

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

Yonghui Xiao, Li Xiong, and Chun Yuan. Differentially private data release through multidimensional partitioning. In *Secure Data Management*, pp. 150–168, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15546-8.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. 2018.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. Privsyn: Differentially private data synthesis. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

Lijia Zhou, Frederic Koehler, Danica J. Sutherland, and Nathan Srebro. Optimistic rates: A unifying theory for interpolation learning and regularization in linear regression, 2021.

T. Zhu, G. Li, W. Zhou, and P. S. Yu. Differentially private data publishing and analysis: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1619–1638, August 2017. ISSN 1041-4347. doi: 10.1109/TKDE.2017.2697856.