Differentially Private Data Generation Needs Better Features

Frederik Harder  
MPI-IS & University of Tübingen

Milad Jalali Asadabadi  
UBC

Danica J. Sutherland  
UBC & Amii

Mijung Park  
UBC & Amii

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.

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 composable, 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 (Y. Xiao et al. 2010; Mohammed et al. 2011; 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; Xie et al. 2018; Frigerio et al. 2019; Torkzadehmohammadi et al. 2019; Yoon 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 and Cortes 2010) or FashionMNIST (H. 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 and Boneh (2021): We simply need better features for differentially private learning. While Tramer and 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 compute 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 Dziugaite et al. 2015, Li et al. 2015, dos Santos et al. 2019, Harder et al. 2021). 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.

2 BACKGROUND

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 $\phi$ maps data in $X$ to a Hilbert space $\mathcal{H}$ (Gretton et al. 2012). One definition is

$$\text{MMD}_{k_\phi}(P, Q) = \left\lVert \mathbb{E}_{x \sim P} \phi(x) - \mathbb{E}_{y \sim Q} \phi(y) \right\rVert_{\mathcal{H}},$$

where $\mu_\phi(P) = \mathbb{E}_{x \sim P} \phi(x) \in \mathcal{H}$ 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 $k_\phi$ is characteristic (Sriperumbudur et al. 2011), then $P \mapsto \mu_\phi(P)$ is injective, so $\text{MMD}_{k_\phi}(P, Q) = 0$ if and only if $P = Q$.

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

$$\text{MMD}_{k_\phi}(\hat{D}, \hat{D}) = \left\lVert \frac{1}{m} \sum_{i=1}^m \phi(x_i) - \frac{1}{n} \sum_{i=1}^n \phi(\tilde{x}_i) \right\rVert_{\mathcal{H}}.$$

3 METHOD

In this paper, to transfer knowledge from public to private data distributions, we construct a particular kernel $k_\phi$ to use in $\text{MMD}_{k_\phi}$ 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-MEPEF), analogous to the related method DP-MERF (Harder et al. 2021).

The relationship can be numerically computed by packages like auto-dp (Wang et al. 2019), among other methods.
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 \( x \), the pre-trained feature extractor network outputs the perceptual features from each layer, where the \( j \)-th layer’s PF is denoted by \( e_j(x) \). Each of the \( J \) layers’ perceptual features is of a different length, \( e_j(x) \in \mathbb{R}^{d_j} \); 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 \( \Phi(x) := [\phi_1(x), \phi_2(x)] \), where the first part comes from a concatenation of PFs from all the layers: \( \phi_1(x) = [e_1(x), \ldots, e_J(x)] \), while the second part comes from their squared values: \( \phi_2(x) = [e_1^2(x), \ldots, e_J^2(x)] \), where \( e_J^2(x) \) means each entry of \( e_J(x) \) is squared. Using this feature map, we then construct the mean embedding of a data distribution given the data samples \( \{x_i\}_{i=1}^m \):

\[
\mu_\Phi(D) = \begin{bmatrix} \mu_{\phi_1}(D) \\ \mu_{\phi_2}(D) \end{bmatrix} = \begin{bmatrix} \frac{1}{m} \sum_{i=1}^m \phi_1(x_i) \\ \frac{1}{m} \sum_{i=1}^m \phi_2(x_i) \end{bmatrix}.
\]

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

\[
\text{Privatization of mean embedding}\quad \text{we privatize the mean embedding of the data distribution only once, and reuse it repeatedly during the training of the generator \( g_\theta \). 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 \( \|\phi_1(x_i)\|_2 = 1 \) and \( \|\phi_2(x_i)\|_2 = 1 \) for each sample \( x_i \). After this change, the}
We now bound the effect of adding noise to our loss function, $\sigma$, where $\mu_{\Phi_i}(D)$ 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

$$
\tilde{\mu}_\Phi(D) = \left[ \tilde{\mu}_{\Phi_1}(D) \right] = \left[ \frac{1}{m} \sum_{i=1}^{m} \phi_1(x_i) + n_1 \right], \quad \left[ \tilde{\mu}_{\Phi_2}(D) \right] = \left[ \frac{1}{m} \sum_{i=1}^{m} \phi_2(x_i) + n_2 \right],
$$

where $n_t \sim \mathcal{N}(0, \frac{4\sigma^2}{m^2} I)$ 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 $\epsilon, \delta$, we use the package of Wang et al. (2019) to find the corresponding $\sigma$ 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 (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: $\sigma_{\text{stopping}} = 10\sigma$, where $\sigma$ is the noise scale for privatizing each part of the data mean embeddings, works well. Again, we compose these $\sigma$s 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 $O_p$ notation for simplicity. $X = O_p(A_n)$ essentially means that $X$ is $O(A_n)$ with probability at least $1 - \rho$ for any constant choice of failure probability $\rho > 0$.

The full version in the supplementary material is also amenable to the choice of covariance for the noise variable $n$, 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 $D$ and $\tilde{D}$, the absolute difference between the privatized and non-private squared MMDs, a random function of only $n$, satisfies

$$
|\text{MMD}_{k_n}(D, \tilde{D}) - \text{MMD}_{k_n}^2(D, \tilde{D})| = O_P \left( \frac{\sigma^2 D}{m} + \frac{\sigma \sqrt{D}}{m} \right).
$$

One key quantity in the bound is $\sigma/m$, the ratio of the noise scale ($\sigma$ inversely proportional to $\epsilon$) to the number of observed (private) data points $m$. Note that $\sigma$ depends only on the given privacy level, not on $m$, so the error becomes zero as long as $m \rightarrow \infty$. In the second term, $\sigma/m$ 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 $D$ features dimensions; although $D$ is typically large, so is $m^2$. Having $m = 50k$ private samples, e.g. for CIFAR-10, allows for a strong error bound as long as $D \sigma^2 \ll 652M$.

The above result is for a fixed pair of datasets. Because we only add noise 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 $D$. For each $\theta$ in some set $\Theta$, fix a corresponding $\hat{D}_\theta$; in particular, $\Theta = \mathbb{R}^p$ could be the set of all generator parameters, and $\hat{D}_\theta$ either the outcome of running a generator $g_\theta$ on a fixed set of “seeds,” $\hat{D}_\theta = \{g_\theta(x_i)\}_{i=1}^m$, or the full output distribution of the generator $Q_{g_\theta}$. Let $\theta \in \arg\min_{\theta \in \Theta} \tilde{\text{MMD}}_{k_n}^2(D, \hat{D}_\theta)$ be the private minimizer, and $\hat{\theta} \in \arg\min_{\theta \in \Theta} \text{MMD}_{k_n}^2(D, \hat{D}_\theta)$ the non-private minimizer. Then

$$
\text{MMD}_{k_n}^2(D, \hat{D}_\theta) - \text{MMD}_{k_n}^2(D, \hat{D}_\theta) = O_P \left( \frac{\sigma^2 D}{m^2} + \frac{\sigma \sqrt{D}}{m} \right).
$$

The second term of this bound will generally dominate; it arises from uniformly bounding the $\frac{\sigma^2}{m} \text{MMD}_{k_n}^2(D, \hat{D}_\theta)$ term of Proposition 4.1 over all possible $\hat{D}_\theta$. This approach, although the default way to prove this type of bound, misses that $\tilde{\text{MMD}}_{k_n}^2(D, \hat{D}_\theta)$ is hopefully small for $\theta$ and $\hat{\theta}$. 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

\[
\text{MMD}_k^2(D, \tilde{D}_\theta) - \text{MMD}_k^2(D, \bar{D}_\theta) = \mathcal{O}(\frac{m^2}{\sigma^2} + \frac{\sqrt{D}}{m} \text{MMD}_k^2(D, \bar{D}_\theta)).
\]

Note that this bound implies the previous one, since \(\text{MMD}_k(D, \bar{D})\) 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 \(\text{MMD}(D, Q_{\theta_0})\) converges to minimizing the true, distribution-level \(\text{MMD}(P, Q_{\theta_0})\): 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 \(1/m^2\) rate.

## 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 (Y. Xiao et al. 2014; Mohammed et al. 2011; Hardt et al. 2012; Zhu et al. 2017; Snoke and Slavkovic 2018). 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 (Qardaji et al. 2014; R. Chen et al. 2015; J. Zhang et al. 2017; Z. 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; Frigerio et al. 2019; Torkzadehmahani et al. 2019; Yoon et al. 2019; D. 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 and Recht 2008). A recent variant of DP-MERF uses Hermite polynomial-based mean embeddings (Vinanzo 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.

**Datasets.** We considered four image datasets of varying complexity. We started with the common datasets MNIST (LeCun and Cortes 2010) and FashionMNIST (H. Xiao et al. 2017), where each consist of 60,000 \(28 \times 28\) 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 \(32 \times 32\) or \(64 \times 64\) pixels and treat as unlabeled. We also study CIFAR-10 (Krizhevsky 2009), a 50,000-sample dataset containing \(32 \times 32\) 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 (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

---

1. Dataset licenses: MNIST: CC BY-SA 3.0; FashionMNIST-MIT: CelebA: see [https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html](https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html); Cifar10: MIT [https://github.com/yuxiangw/autodp](https://github.com/yuxiangw/autodp)
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 $\phi_1$, $\phi_2$.

|               | DP-MEPF | DP-Sinkhorn | GS-WGAN | DP-MERF | DP-HP |
|---------------|---------|-------------|---------|---------|-------|
|               | LogReg  | MLP         | LogReg  | MLP     |       |
| MNIST         | 83      | 90          | 83      | 79      | 79    | 81    |
| F-MNIST       | 76      | 75          | 68      | 76      | 75    | 73    |

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 and 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 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 Torkzadehmahani et al. 2019, Yoon et al. 2019, D. Chen et al. 2020, Cao et al. 2021, Harder 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 (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 models based on embeddings with only the first moment, written $(\phi_1)$, and using the first two moments, written $(\phi_1, \phi_2)$. 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 baseline based on public data, we also pretrain a GAN on a downscaled version of ImageNet, at $32 \times 32$, and fine-tune this model with DP-SGD on CelebA and Cifar10. We use architectures based on ResNet9 with group normalization (Wu and 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.

**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 2 shows that downstream test accuracy only starts to drop in high privacy regimes, $\epsilon < 1$, due to the low sensitivity of $\mu_0$. 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 sam-
samples at the same $32 \times 32$ 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 $64 \times 64$ images are shown in Figure 3 with similar quality; Table 3 gives FID scores for both resolutions at varying $\epsilon$.

For CelebA and Cifar10, we use DP early stopping as explained in Section 3.2 with a privacy parameter ten times larger than the $\sigma$ used for the training objective. Keeping $(\epsilon, \delta)$ fixed, this additional release results only in a small increase in $\sigma$, 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.

**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.

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

| Privacy | DP-MEPF ($\phi_1$) | DP-MEPF ($\phi_1, \phi_2$) | DP-MERF |
|---------|---------------------|----------------------------|---------|
| $\epsilon = 10$ | 51.0 | 48.9 | 13.2 |
| $\epsilon = 5$ | 48.5 | 47.9 | 13.4 |
| $\epsilon = 2$ | 42.5 | 38.7 | 13.5 |
| $\epsilon = 1$ | 39.4 | 28.9 | 13.8 |
| $\epsilon = 0.5$ | 19.4 | 19.7 | 13.1 |
| $\epsilon = 0.2$ | 13.8 | 12.4 | 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 $\mu_\phi$ 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., $\epsilon \leq 2$. 
Sriperumbudur, B. K., K. Fukumizu, and G. R. Lanckriet (2011). “Universality, Characteristic Kernels and RKHS Embedding of Measures.” Journal of Machine Learning Research 12.7.

Torkzadehmahani, R., P. Kairouz, and B. Paten (June 2019). “DP-CGAN: Differentially Private Synthetic Data and Label Generation.” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

Tramer, F. and D. Boneh (2021). “Differentially Private Learning Needs Better Features (or Much More Data).” International Conference on Learning Representations.

Vinaroz, M., M.-A. Charusaisa, F. Harder, K. Adamczewski, and M. J. Park (2022). “Hermite Polynomial Features for Private Data Generation.” ICML. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 22300–22324.

Wang, Y.-X., B. Bale, and S. P. Kasiviswanathan (2019). “Subsampled Rényi differential privacy and analytical moments accountant.” AISTATS.

Wu, Y. and K. He (2018). “Group normalization.” Proceedings of the European conference on computer vision (ECCV), pp. 3–19.

Xiao, H., K. Rasul, and R. Vollgraf (2017). “Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms.” arXiv: [1708.07747].

Xiao, Y., L. Xiong, and C. Yuan (2010). “Differentially Private Data Release through Multidimensional Partitioning.” Secure Data Management. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 150–168.

Xie, L., K. Lin, S. Wang, F. Wang, and J. Zhou (2018). “Differentially Private Generative Adversarial Network.” arXiv: [1802.06739].

Yoon, J., J. Jordon, and M. van der Schaar (2019). “PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees.” International Conference on Learning Representations.

Zhang, J., G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao (2017). “Privbayes: Private data release via bayesian networks.” ACM Transactions on Database Systems (TODS) 42.4, pp. 1–41.

Zhang, Z., T. Wang, N. Li, J. Honorio, M. Backes, S. He, J. Chen, and Y. Zhang (2021). “Privsyn: Differentially private data synthesis.” 30th USENIX Security Symposium (USENIX Security 21).

Zhou, L., F. Koehler, D. J. Sutherland, and N. Srebro (2021). Optimistic Rates: A Unifying Theory for Interpolation Learning and Regularization in Linear Regression. arXiv: [2112.04470].

Zhu, T., G. Li, W. Zhou, and P. S. Yu (Aug. 2017). “Differentially Private Data Publishing and Analysis: A Survey.” IEEE Transactions on Knowledge and Data Engineering 29.8, pp. 1619–1638.

differential-privacy-for-census-data-explained.aspx

Papernot, N., M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar (2017). “Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data.” Proceedings of the International Conference on Learning Representations (ICLR). arXiv: [1610.05755]

Park, M., J. Foulds, K. Choudhary, and M. Welling (Apr. 2017). “DP-EM: Differentially Private Expectation Maximization.” AISTATS. Vol. 54. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, pp. 896–904.

Paszke, A. et al. (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library.” Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035.

Pedregosa, F., 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 (2011). “Scikit-learn: Machine Learning in Python.” Journal of Machine Learning Research 12, pp. 2825–2830.

Qardaji, W., W. Yang, and N. Li (2014). “Privview: practical differentially private release of marginal contingency tables.” Proceedings of the 2014 ACM SIGMOD international conference on Management of data, pp. 1435–1446.

Rahimi, A. and B. Recht (2008). “Random features for large-scale kernel machines.” Advances in Neural Information Processing Systems, pp. 1177–1184.

Seitzer, M. (Aug. 2020). pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid. Version 0.2.1.

Simonyan, K. and A. Zisserman (2014). “Very deep convolutional networks for large-scale image recognition.” arXiv: [1409.1556].

Smola, A., A. Gretton, L. Song, and B. Schölkopf (2007). “A Hilbert space embedding for distributions.” ALT, pp. 13–31.

Snoke, J. and A. Slavković (2018). “pMSE mechanism: differentially private synthetic data with maximal distributional similarity.” International Conference on Privacy in Statistical Databases. Springer, pp. 138–159.

Song, L., K. Fukumizu, and A. Gretton (2013). “Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models.” IEEE Signal Processing Magazine 30.4, pp. 98–111.

Srebro, N., K. Sridharan, and A. Tewari (2010). Optimistic Rates for Learning with a Smooth Loss. arXiv: [1009.3896].

Sriperumbudur, B. K., K. Fukumizu, and G. R. Lanckriet (2011). “Universality, Characteristic Kernels and RKHS Embedding of Measures.” Journal of Machine Learning Research 12.7.
A Proofs

We will conduct our analysis in terms of general noise covariance $\Sigma$ for the added noise, $n \sim \mathcal{N}(0, \Sigma)$. The results will depend on various norms of $\Sigma$, as well as $\|\Sigma^{1/2}a\|$, where $a = \mu_\phi(D) - \mu_\phi(\tilde{D})$ is the difference between empirical mean embeddings $\mu_\phi(D) = \frac{1}{m}\sum_{x \in D} \phi(x)$. (Recall that $\text{MMD}(D, \tilde{D}) = \|a\|$.) When we use only normalized first-moment features, the quantities appearing in the bounds are

$$\Sigma = \frac{4\sigma^2}{m^2} I_D$$

$$\|\Sigma\|_{op} = \frac{4\sigma^2}{m^2} \quad \|\Sigma\|_F = \frac{4\sigma^2}{m^2} \sqrt{D} \quad \text{Tr}(\Sigma) = \frac{4\sigma^2}{m^2} D$$

(6)

When we use first- and second-moment features with respective scales $C_1$ and $C_2$ (both 1 in our experiments here), we have

$$\Sigma = \begin{bmatrix}
\sigma^2 \left(\frac{2C_1^2}{m^2}\right) I_D & 0 \\
0 & \sigma^2 \left(\frac{2C_2^2}{m^2}\right) I_D
\end{bmatrix} = \frac{4\sigma^2}{m^2} \begin{bmatrix}
C_1^2 I_D & 0 \\
0 & C_2^2 I_D
\end{bmatrix}$$

$$\|\Sigma\|_{op} = \frac{4\sigma^2}{m^2} \max(C_1^2, C_2^2) \quad \|\Sigma\|_F = \frac{4\sigma^2}{m^2} (C_1^2 + C_2^2) \sqrt{D} \quad \text{Tr}(\Sigma) = \frac{4\sigma^2}{m^2} (C_1^2 + C_2^2) D$$

(7)

$$\|\Sigma^{1/2}a\|_2 = \sqrt{a^\top \Sigma a} = \frac{2\sigma}{m} \sqrt{C_1^2 \text{MMD} k_{\phi_1}(D, \tilde{D})^2 + C_2^2 \text{MMD} k_{\phi_2}(D, \tilde{D})^2}.$$

Note that if $C_1 = C_2 = C$, then

$$\sqrt{C_1^2 \text{MMD} k_{\phi_1}(D, \tilde{D})^2 + C_2^2 \text{MMD} k_{\phi_2}(D, \tilde{D})^2} = C \text{MMD} k_{\phi_2}(D, \tilde{D}).$$

A.1 Mean absolute error of loss function

**Proposition A.1.** Given datasets $D = \{x_i\}_{i=1}^m$ and $\tilde{D} = \{\tilde{x}_j\}_{j=1}^m$ and a kernel $k_\phi$ with a $D$-dimensional embedding $\phi$, let $a = \mu_\phi(D) - \mu_\phi(\tilde{D})$. Define $\text{MMD}_{k_\phi}(D, \tilde{D}) = \|a + n\|^2$ for a noise vector $n \sim \mathcal{N}(0, \Sigma)$. Introducing the noise $n$ affects the expected absolute error as

$$\mathbb{E}_n \left[ |\text{MMD}_{k_\phi}(D, \tilde{D}) - \text{MMD}_{k_\phi}(D, \tilde{D})|^2 \right] \leq \text{Tr}(\Sigma) + 2 \sqrt{\frac{2}{\pi} \|\Sigma^{1/2}a\|}.$$ 

(8)

**Proof.** We have that

$$\mathbb{E}_n \left[ |\text{MMD}_{k_\phi}(D, \tilde{D}) - \text{MMD}_{k_\phi}(D, \tilde{D})|^2 \right] = \mathbb{E}_n \left[ \|a + n\|^2 - \|a\|^2 \right] = \mathbb{E}_n \left[ n^\top n + 2n^\top a \right] \leq \mathbb{E}_n \left[ n^\top n \right] + 2 \mathbb{E}_n \left[ n^\top a \right].$$

(9)

The first term is standard:

$$\mathbb{E}_n n^\top n = \mathbb{E} \text{Tr}(n^\top n) = \mathbb{E} \text{Tr}(nn^\top) = \text{Tr}(\mathbb{E}nn^\top) = \text{Tr}(\Sigma).$$

For the second, note that

$$a^\top n \sim \mathcal{N}(0, a^\top \Sigma a),$$

and so its absolute value is $\sqrt{a^\top \Sigma a}$ times a $\chi(1)$ random variable. Since the mean of a $\chi(1)$ distribution is $\sqrt{\frac{2}{\pi}} \Gamma(1) = \sqrt{\frac{2}{\pi}}$, we obtain the desired bound.
A.2 High-probability bound on the error

**Proposition A.2.** Given datasets $\mathcal{D} = \{x_i\}_{i=1}^n$ and $\mathcal{D} = \{\hat{x}_j\}_{j=1}^m$, let $a = \mu_\phi(\mathcal{D}) - \mu_\phi(\hat{\mathcal{D}})$, and define $\tilde{\text{MMD}}_{k_0}^2(\mathcal{D}, \hat{\mathcal{D}}) = ||a + n||^2$ for a noise vector $n \sim \mathcal{N}(0, \Sigma)$. Then for any $\rho \in (0, 1)$, it holds with probability at least $1 - \rho$ over the choice of $n$ that

\[
\frac{\tilde{\text{MMD}}_{k_0}^2(\mathcal{D}, \hat{\mathcal{D}}) - \text{MMD}_{k_0}^2(\mathcal{D}, \hat{\mathcal{D}})}{||a + n||^2} \leq \text{Tr}(\Sigma) + \sqrt{2||\Sigma^2a||_2} + 2 \left( ||\Sigma||_F + \sqrt{2||\Sigma^2a||_2} \right) \sqrt{\log(\frac{2}{\rho})} + 2||\Sigma||_{op} \log(\frac{2}{\rho}).
\]

This implies that

\[
\frac{\tilde{\text{MMD}}_{k_0}^2(\mathcal{D}, \hat{\mathcal{D}}) - \text{MMD}_{k_0}^2(\mathcal{D}, \hat{\mathcal{D}})}{||a + n||^2} = \mathcal{O}_P \left( \text{Tr}(\Sigma) + ||\Sigma^{1/2}a||_2 \right).
\]

**Proof.** Introduce $z \sim \mathcal{N}(0, I)$ such that $n = \Sigma^{1/2}z$ into (10):

\[
\left| \frac{\tilde{\text{MMD}}_{k_0}^2(\mathcal{D}, \hat{\mathcal{D}}) - \text{MMD}_{k_0}^2(\mathcal{D}, \hat{\mathcal{D}})}{||a + n||^2} \right| \leq n^\top a + 2||n^\top a|| = z^\top \Sigma z + 2|a^\top \Sigma^{1/2}z|.
\]

For the first term, denoting the eigendecomposition of $\Sigma$ as $QAQ^\top$, we can write

\[
z^\top \Sigma z = (Q^\top z)^\top \Lambda (Q^\top z),
\]

in which $Q^\top z \sim \mathcal{N}(0, I)$ and $\Lambda$ is diagonal. Thus, applying Lemma 1 of Laurent and Massart (2000), we obtain that with probability at least $1 - \frac{\rho}{2}$,

\[
z^\top \Sigma z \leq \text{Tr}(\Sigma) + 2||\Sigma^1_2|| \sqrt{\log(\frac{2}{\rho})} + 2||\Sigma^1||_{op} \log(\frac{2}{\rho}).
\]

In the second term, $|a^\top \Sigma^{1/2}z|$, can be viewed as a function of a standard normal variable $z$ with Lipschitz constant at most $||\Sigma^{1/2}a||_2$. Thus, applying the standard Gaussian Lipschitz concentration inequality (Boucheron et al. 2013 Theorem 5.6), we obtain that with probability at least $1 - \frac{\rho}{2}$,

\[
|z^\top \Sigma^{1/2}a| \leq \mathbb{E}|z^\top \Sigma^{1/2}a| + ||\Sigma^{1/2}a||_2 \sqrt{2\log(\frac{2}{\rho})} = ||\Sigma^{1/2}a||_2 \left( \sqrt{\frac{2}{\rho}} + \sqrt{2\log(\frac{2}{\rho})} \right).
\]

The first statement in the theorem follows by a union bound. The $\mathcal{O}_P$ form follows by Lemma A.1 and the fact that $\text{Tr}(A) \geq ||A||_F \geq ||A||_{op}$ for positive semi-definite matrices $A$.

The following lemma shows how to convert high-probability bounds with both sub-exponential and sub-Gaussian tails into a $\mathcal{O}_P$ statement.

**Lemma A.1.** If a sequence of random variables $X_n$ satisfies

\[
X_n \leq A_n + B_n \sqrt{\log \frac{b_n}{\rho}} + C_n \log \frac{c_n}{\rho} \quad \text{with probability at least } 1 - \rho,
\]

then the sequence of variables $X_n$ is

\[
\mathcal{O}_P \left( \max \left( A_n, B_n \max(\sqrt{\log b_n}, 1), C_n \max(\log c_n, 1) \right) \right).
\]

**Proof.** The definition of a sequence of random variables $X_n$ being $\mathcal{O}_P(Q_n)$, where $Q_n$ is a sequence of scalars, means that the sequence $\frac{X_n}{Q_n}$ is stochastically bounded: for each $\rho$, there is some constant $R_\rho$ such that $\Pr(X_n/Q_n \geq R_\rho) \leq \rho$. Here,
we have for all \( n \) with probability at least \( 1 - \rho \) that

\[
X_n \leq \frac{A_n + B_n \sqrt{\log \frac{b_n}{\rho} + C_n \log \frac{b_n}{\rho}}}{\max (A_n, B_n \max (\sqrt{\log b_n}, 1), C_n \max (\log c_n, 1))}
\]

Thus the desired bound holds with \( R_{\rho} = 3 + \sqrt{\log \frac{1}{\rho} + \log \frac{1}{\rho}}. \)

A.3 Quality of the private minimizer: worst-case analysis

We first show uniform convergence of the privatized MMD to the non-private MMD.

**Proposition A.3.** Suppose that \( \Phi : X \rightarrow \mathbb{R}^D \) is such that \( \sup_x \| \Phi(x) \| \leq B \), and let \( \overline{\text{MMD}}_{k_b}(D, \tilde{D}) = \| \mu_\Phi(D) - \mu_\Phi(\tilde{D}) + n \| \) for \( n \sim \mathcal{N}(0, \Sigma) \). Then, with probability at least \( 1 - \rho \) over the choice of \( n \),

\[
\sup_{D, \tilde{D}} \left| \overline{\text{MMD}}_{k_b}^2(D, \tilde{D}) - \text{MMD}_{k_b}^2(D, \tilde{D}) \right| \leq \text{Tr}(\Sigma) + 4B \sqrt{\text{Tr}(\Sigma)} + 2 \left( \| \Sigma \|_F + 2B \| \Sigma \|_{op} \right) \sqrt{\log \left( \frac{2}{\rho} \right) + 2 \| \Sigma \|_{op} \log \left( \frac{2}{\rho} \right)}
\]

where the supremum is taken over all probability distributions, including the empirical distribution of datasets \( D, \tilde{D} \) of any size.

**Proof.** Introducing \( z \sim \mathcal{N}(0, I_D) \) such that \( n = \Sigma^{1/2} z \), we have that

\[
\sup_{D, \tilde{D}} \left| \overline{\text{MMD}}_{k_b}^2(D, \tilde{D}) - \text{MMD}_{k_b}^2(D, \tilde{D}) \right| \leq \sup_{D, \tilde{D}} z^\top \Sigma z + 2 \| a \| \Sigma^{1/2} z \|
\]

To apply Gaussian Lipschitz concentration, we also need to know that

\[
\mathbb{E}\| \Sigma^{1/2} z \| \leq \sqrt{\mathbb{E}\| \Sigma^{1/2} z \|^2} = \sqrt{\text{Tr}(\Sigma)};
\]

the exact expectation of a \( \chi \) variable with more than one degree of freedom is inconvenient, but the gap is generally not asymptotically significant. Then we get that, with probability at least \( 1 - \rho \),

\[
\| \Sigma^{1/2} z \| \leq \sqrt{\text{Tr}(\Sigma)} + \| \Sigma \|_{op} \sqrt{2 \log \frac{2}{\rho}}.
\]

Again combining with the bound of [12], we get the stated bound.
This bound is looser than in Proposition A.2, since the term depending on $a$ is now “looking at” $z$ in many directions rather than just one: we end up with a $\chi(\dim(\Sigma))$ random variable instead of $\chi(1)$.

We can use this uniform convergence bound to show that the minimizer of the private loss approximately minimizes the non-private loss:

**Proposition A.4.** Fix a target dataset $D$. For each $\theta$ in some set $\Theta$, fix a corresponding $\tilde{D}_\theta$; in particular, $\Theta = \mathbb{R}^p$ could be the set of all generator parameters, and $\tilde{D}_\theta$ either the outcome of running a generator $g_\theta$ on a fixed set of “seeds,” $\tilde{D}_\theta = \{g_\theta(z_i)\}_{i=1}^n$, or the full output distribution of the generator $Q_{g_\theta}$. Suppose that $\Phi: \mathcal{X} \to \mathbb{R}^D$ is such that $\sup_x \|\Phi(x)\| \leq B$, and let $\text{MMD}_{k_\Phi}(D, \tilde{D}) = \|\mu_\Phi(D) - \mu_\Phi(\tilde{D}) + n\|$ for $n \sim N(0, \Sigma)$. Let $\tilde{\theta} \in \arg\min_{\theta \in \Theta} \text{MMD}_{k_\Phi}(D, \tilde{D}_\theta)$ be the private minimizer, and $\hat{\theta} \in \arg\min_{\theta \in \Theta} \text{MMD}_{k_\Phi}(D, \tilde{D}_\theta)$ the non-private minimizer. For any $\rho \in (0, 1)$, with probability at least $1 - \rho$ over the choice of $n$,

$$\text{MMD}_{k_\Phi}^2(D, \tilde{D}_\theta) - \text{MMD}_{k_\Phi}^2(D, \tilde{D}_\theta) \leq 2\text{Tr}(\Sigma) + 8B\sqrt{\text{Tr}(\Sigma)} + 4\left(\|\Sigma\|_F + 2B\|\Sigma\|_{op}\right)\sqrt{\log(\frac{1}{\rho})} + 4\|\Sigma\|_{op}\log(\frac{1}{\rho}) = O_P\left(\text{Tr}(\Sigma) + B\sqrt{\text{Tr}(\Sigma)}\right).$$

**Proof.** Let $\alpha$ represent the uniform error bound of Proposition A.2. Applying Proposition A.2, the definition of $\tilde{\theta}$, then Proposition A.2 again:

$$\text{MMD}_{k_\Phi}^2(D, \tilde{D}_\theta) \leq \text{MMD}_{k_\Phi}^2(D, \tilde{D}_\theta) + \alpha \leq \text{MMD}_{k_\Phi}^2(D, \tilde{D}_\theta) + \alpha \leq \text{MMD}_{k_\Phi}^2(D, \tilde{D}_\theta) + 2\alpha.$$

A.4 Quality of the private minimizer: “optimistic” analysis

The preceding analysis is quite “worst-case,” since we upper-bounded the MMD by the maximum possible value everywhere. Noticing that the approximation in Proposition A.2 is tighter when $\|\Sigma^{1/2}a\|$ is smaller, we can instead show an “optimistic” rate which takes advantage of this fact to show tighter approximation for the minimizer of the noised loss. In the “interpolating” case where the generator can achieve zero empirical MMD, the convergence rate substantially improves (generally improving the squared MMD from $O_P(1/m)$ to $O_P(1/m^2)$).

**Proposition A.5.** In the setup of Proposition A.4, we have with probability at least $1 - \rho$ over $n$ that

$$\text{MMD}_{k_\Phi}^2(D, \tilde{D}_\theta) - \text{MMD}_{k_\Phi}^2(D, \tilde{D}_\theta) \leq 9\text{Tr}(\Sigma) + 4\sqrt{\text{Tr}(\Sigma)} \text{MMD}_{k_\Phi}(D, \tilde{D}_\theta) + 2\left(9\|\Sigma\|_F + 2\|\Sigma\|_{op}\text{MMD}_{k_\Phi}(D, \tilde{D}_\theta)\right)\sqrt{\log(\frac{2}{\rho})} + 18\|\Sigma\|_{op}\log(\frac{2}{\rho}) = O_P\left(\text{Tr}(\Sigma) + \sqrt{\text{Tr}(\Sigma)} \text{MMD}_{k_\Phi}(D, \tilde{D}_\theta)\right).$$

**Proof.** Let’s use $\tilde{\text{MMD}}(\theta)$ to denote $\text{MMD}_{k_\Phi}(D, \tilde{D}_\theta)$, and $\hat{\text{MMD}}(\theta)$ for $\text{MMD}_{k_\Phi}(D, \tilde{D}_\theta)$.

For all $\theta$, we have that

$$\left|\tilde{\text{MMD}}^2(\theta) - \hat{\text{MMD}}^2(\theta)\right| \leq z^T \Sigma z + 2(\mu_\Phi(D) - \mu_\Phi(\tilde{D}))^T \Sigma^{1/2} z \leq z^T \Sigma z + 2 \tilde{\text{MMD}}(\theta)\|\Sigma^{1/2} z\|.$$

Thus, applying this inequality in both the first and third lines,

$$\tilde{\text{MMD}}^2(\theta) \leq \hat{\text{MMD}}^2(\theta) + z^T \Sigma z + 2 \tilde{\text{MMD}}(\theta)\|\Sigma^{1/2} z\| \leq \text{MMD}^2(\theta) + z^T \Sigma z + 2 \text{MMD}(\theta)\|\Sigma^{1/2} z\| \leq \text{MMD}^2(\theta) + 2z^T \Sigma z + 2 \left(\text{MMD}(\theta) + \tilde{\text{MMD}}(\theta)\right)\|\Sigma^{1/2} z\|.$$
in the second line we used that $\widehat{\text{MMD}}(\bar{\theta}) \leq \widehat{\text{MMD}}(\hat{\theta})$. Rearranging, we get that
\[
\widehat{\text{MMD}}^2(\hat{\theta}) - \beta \widehat{\text{MMD}}(\hat{\theta}) - \gamma \leq 0,
\]
where
\[
\beta = 2\|\Sigma^{1/2}z\| \geq 0
\]
\[
\gamma = \widehat{\text{MMD}}^2(\hat{\theta}) + 2z^T \Sigma z + 2 \widehat{\text{MMD}}(\hat{\theta})\|\Sigma^{1/2}z\| \geq 0.
\]
The left-hand side of (13) is a quadratic in $\widehat{\text{MMD}}(\hat{\theta})$ with positive curvature; it has two roots, at
\[
\frac{\beta}{2} \pm \sqrt{\left(\frac{\beta}{2}\right)^2 + \gamma}.
\]
Thus the inequality (13) can only hold in between the roots; the root with a minus sign is negative, and so does not concern us since we know that $\text{MMD}(\hat{\theta}) \geq 0$. Thus, for (13) to hold, we must have
\[
\widehat{\text{MMD}}(\hat{\theta}) \leq \frac{\beta}{2} + \sqrt{\left(\frac{\beta}{2}\right)^2 + \gamma}
\]
\[
\widehat{\text{MMD}}^2(\hat{\theta}) \leq \frac{\beta^2}{4} + \left(\frac{\beta}{2}\right)^2 + \gamma + \beta\sqrt{\left(\frac{\beta}{2}\right)^2 + \gamma}
\]
\[
\leq \gamma + \beta^2 + \beta\sqrt{\gamma}.
\]
Also note that
\[
\gamma = \widehat{\text{MMD}}^2(\hat{\theta}) + 2z^T \Sigma z + 2 \widehat{\text{MMD}}(\hat{\theta})\|\Sigma^{1/2}z\| \leq \left(\widehat{\text{MMD}}(\hat{\theta}) + \sqrt{2}\|\Sigma^{1/2}z\|\right)^2.
\]
Thus, substituting in for $\beta$ and $\gamma$ then simplifying, we have that
\[
\widehat{\text{MMD}}^2(\hat{\theta}) \leq \widehat{\text{MMD}}^2(\hat{\theta}) + (6 + 2\sqrt{2})z^T \Sigma z + 4\|\Sigma^{1/2}z\| \widehat{\text{MMD}}(\hat{\theta}).
\]
Using the same bounds on $z^T \Sigma z$ and $\|\Sigma^{1/2}z\|$ as in Proposition A.3 and $6\sqrt{2} < 9$, gives the claimed bound.

\section{Extended Implementation details}

\textbf{Repository.} Our anonymized code is available at \url{https://anonymous.4open.science/r/dp-gfmn}; the readme files contain further instructions on how to run the code.

\subsection{Hyperparameter settings}

For each dataset, we tune the generator learning rate ($\text{LR}_{\text{gen}}$) and moving average learning rate ($\text{LR}_{\text{mavg}}$) from choices $10^{-k}$ and $3 \cdot 10^{-k}$ with $k \in \{3, 4, 5\}$ once for the non-private setting and once at $\epsilon = 2$. The latter is used in all private experiments for that dataset, as shown in [7]. After some initial unstructured experimentation, hyperparameters are chosen with identical values across dataset shown in [8].

For the Cifar10 DP-MERF baseline we tested random tuned random features dimension $d \in \{10000, 50000\}$, random features sampling distribution $\sigma \in \{100, 300, 1000\}$, learning rate decay by 10% every $e \in \{1000, 10000\}$ iterations and learning rate $10^{-k}$ with $k \in \{2, 3, 4, 5, 6\}$. Results presented use $d = 500000, \sigma = 1000, e = 10000, k = 3$.

The DP-GAN baseline for Cifar10 and CelebA uses the same generator as DP-MEPF with 3 residual blocks and a total of 8 convolutional layers and is paired with a ResNet9 discriminator which uses Groupnorm instead of Batchnorm to allow for per-sample gradient computation. We pre-train the model non-privately to convergence on downsampled imagenet in order to maintain the same resolution of $32 \times 32$ and then fine-tune the model for a smaller number of epochs. Results are the best scores of a gridsearch over the following parameters at $\epsilon = 2$, which is then used in all settings: number of epochs $\{1, 10, 30, 50\}$ generator and discriminator learning rate separately for $10^{-k}$ and $3 \cdot 10^{-k}$ with $k \in \{3, 4, 5\}$, clip-norm $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$, batch size $\{128, 256, 512\}$ and, as advised in Anonymous (2023), number of discriminator updates per generator $\{1, 10, 30, 50\}$. The chosen values are given in table 9.
Table 7: Learning rate hyperparameters across datasets

| Dataset          | $\text{LR}_{\text{gen}}$ | $\text{LR}_{\text{mavg}}$ |
|------------------|---------------------------|---------------------------|
| MNIST-nonDP      | $10^{-5}$                 | $10^{-5}$                 |
| MNIST-DP         | $10^{-5}$                 | $10^{-4}$                 |
| FashionMNIST-nonDP | $10^{-5}$            | $10^{-3}$                 |
| FashionMNIST-DP  | $10^{-4}$                 | $10^{-3}$                 |
| CelebA32-nonDP   | $10^{-3}$ · $10^{-4}$     | $10^{-4}$                 |
| CelebA32-DP      | $10^{-3}$ · $10^{-4}$     |                           |
| CelebA64-nonDP   | $10^{-4}$ · $3 \cdot 10^{-4}$ | $10^{-4}$ · $3 \cdot 10^{-4}$ |
| CelebA64-DP      | $10^{-4}$ · $3 \cdot 10^{-4}$ |                           |
| Cifar10-nonDP labeled | $10^{-3}$ · $10^{-2}$       | $10^{-2}$                 |
| Cifar10-DP labeled | $10^{-3}$ · $10^{-2}$       |                           |
| Cifar10-nonDP unlabeled | $10^{-3}$ · $3 \cdot 10^{-4}$ | $10^{-4}$                 |
| Cifar10-DP unlabeled | $10^{-3}$ · $3 \cdot 10^{-4}$ |                           |

Table 8: Hyperparameters fixed across datasets

| Parameter                              | Value                   |
|----------------------------------------|-------------------------|
| $(\phi_1)$-bound                       | 1                       |
| $(\phi_2)$-bound                       | 1                       |
| iterations (MNIST & FashionMNIST)      | 100,000                 |
| batch size (MNIST and FashionMNIST)    | 100                     |
| iterations (Cifar10 & CelebA)          | 200,000                 |
| batch size (Cifar10 and CelebA)        | 128                     |
| seeds                                  | 1,2,3,4,5               |

C Detailed Tables

Below we present the results from the main paper with added $a \pm b$ notation, where $a$ is the mean and $b$ is the standard deviation of the score distribution across three independent runs for MNIST and FashionMNIST and 5 independent runs for Cifar10 and CelebA.

D Encoder architecture comparison

We are testing a large collection of classifiers of different sizes from the torchvision library including VGG, ResNet, ConvNext and EfficientNet. For each we look at unlabelled Cifar10 generation quality in the non-DP setting and at $\epsilon = 0.2$. In each architecture, we use all activations from convolutional layers with a kernel size greater than 1x1. We list the number of extracted features along with the achieved FID score in table 17, where each result is the best result obtained by tuning learning rates. As already observed in dos Santos et al. (2019), we find that VGG architectures appear to learn particularly useful features for feature matching. We hypothesized that in the private setting other architectures with fewer features might outperform the VGG model, but have found this to not be the case.

E Public dataset comparison

We pretrained a ResNet18 using ImageNet, CIFAR10, and SVHN as our public data, respectively. We then used the perceptual features to train a generator using CelebA dataset as our private data at a privacy budget of $\epsilon = 0.2$ and obtained

Table 9: Hyperparameters of DP-GAN for Cifar10 and CelebA

| Dataset | Cifar10 | CelebA |
|---------|---------|--------|
| $\text{LR}_{\text{gen}}$ | $10^{-4}$ | $3 \cdot 10^{-4}$ |
| $\text{LR}_{\text{dis}}$  | $10^{-3}$ | $3 \cdot 10^{-4}$ |
| batch size | 512     | 512    |
| epochs     | 10       | 10     |
| discriminator frequency  | 10       | 10     |
| clip norm  | $10^{-5}$ | $10^{-4}$ |
Table 10: Downstream accuracies of our method for MNIST at varying values of $\epsilon$

| $\epsilon$ | MLP-MEPF $(\phi_1, \phi_2)$ | MLP-MEPF $(\phi_1)$ | LogReg-MEPF $(\phi_1, \phi_2)$ | LogReg-MEPF $(\phi_1)$ |
|------------|----------------------------|---------------------|--------------------------------|---------------------|
| $\infty$   | 91.4 ± 0.3                 | 88.2 ± 0.6          | 84.6 ± 0.5                     | 81.4 ± 0.4          |
| 10          | 89.8 ± 0.5                 | 88.8 ± 0.1          | 83.4 ± 0.6                     | 80.8 ± 0.9          |
| 5           | 89.9 ± 0.2                 | 88.4 ± 0.5          | 83.3 ± 0.7                     | 80.8 ± 0.8          |
| 2           | 89.3 ± 0.3                 | 88.0 ± 0.2          | 82.9 ± 0.7                     | 80.5 ± 0.6          |
| 1           | 89.3 ± 0.6                 | 87.5 ± 0.6          | 82.5 ± 0.5                     | 79.0 ± 0.6          |
| 0.2         | 79.9 ± 1.3                 | 77.1 ± 0.4          | 75.8 ± 1.1                     | 72.1 ± 1.4          |
| 0.5         | 79.9 ± 0.7                 | 77.1 ± 0.8          | 75.8 ± 1.1                     | 72.1 ± 1.4          |

Table 11: Downstream accuracies of our method for FashionMNIST at varying values of $\epsilon$

| $\epsilon$ | MLP-MEPF $(\phi_1, \phi_2)$ | MLP-MEPF $(\phi_1)$ | LogReg-MEPF $(\phi_1, \phi_2)$ | LogReg-MEPF $(\phi_1)$ |
|------------|----------------------------|---------------------|--------------------------------|---------------------|
| $\infty$   | 74.4 ± 0.3                 | 73.8 ± 0.5          | 74.3 ± 0.1                     | 72.8 ± 0.5          |
| 10          | 76.0 ± 0.4                 | 75.5 ± 0.6          | 75.7 ± 1.0                     | 75.5 ± 0.1          |
| 5           | 75.8 ± 0.6                 | 75.1 ± 0.6          | 75.2 ± 0.4                     | 75.5 ± 0.8          |
| 2           | 75.1 ± 0.3                 | 75.8 ± 0.7          | 75.8 ± 0.4                     | 76.4 ± 0.8          |
| 1           | 74.7 ± 1.1                 | 75.0 ± 1.8          | 75.4 ± 1.1                     | 76.2 ± 0.8          |
| 0.2         | 70.4 ± 1.9                 | 69.0 ± 1.5          | 72.5 ± 1.2                     | 71.7 ± 0.4          |
| 0.5         | 76.4 ± 2.5                 | 76.2 ± 1.4          | 72.5 ± 1.2                     | 71.7 ± 0.4          |

The scores shown in [18]. These numbers reflect our intuition that as long as the public data is sufficiently similar and contains more complex patterns than private data, e.g., transferring the knowledge learned from ImageNet as public data to generate CelebA images as private data, the learned features from public data are useful enough to generate good synthetic data. In addition, as the public data become more simplistic (from CIFAR10 to SVHN), the usefulness of such features reduces in producing good CelebA synthetic samples.

Table 12: CelebA FID scores 32 × 32 (lower is better)

| $\epsilon$ | DP-MEPF $(\phi_1, \phi_2)$ | DP-MEPF $(\phi_1)$ |
|------------|-----------------------------|---------------------|
| $\infty$   | 13.9 ± 1.6                  | 12.8 ±               |
| 10          | 15.1 ± 4.7                  | 11.7 ± 0.6          |
| 5           | 14.3 ± 2.3                  | 12.1 ± 1.1          |
| 2           | 13.9 ± 1.1                  | 12.6 ± 1.0          |
| 1           | 14.9 ± 2.5                  | 13.2 ± 1.6          |
| 0.5         | 14.4 ± 1.7                  | 14.4 ± 1.1          |
| 0.2         | 19.3 ± 3.0                  | 18.1 ± 2.3          |
Table 13: CelebA FID scores 64 × 64 (lower is better)

| $\epsilon$ = $\infty$ | $\epsilon$ = 10 | $\epsilon$ = 5 | $\epsilon$ = 2 | $\epsilon$ = 1 | $\epsilon$ = 0.5 | $\epsilon$ = 0.2 |
|------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| DP-MEPF ($\phi_1, \phi_2$) | 12.8 ± 0.6 | 13.0 ± 0.7 | 13.1 ± 0.9 | 13.2 ± 0.3 | 13.5 ± 1.1 | 15.5 ± 1.0 | 24.8 ± 1.6 |
| DP-MEPF ($\phi_1$) | 11.2 ± 0.5 | 11.7 ± 0.7 | 11.7 ± 0.6 | 11.6 ± 0.4 | 13.0 ± 0.7 | 16.2 ± 0.7 | 27.3 ± 2.3 |

Table 14: FID scores for synthetic labelled CIFAR-10 data (generating both labels and input images)

| $\epsilon$ = $\infty$ | $\epsilon$ = 10 | $\epsilon$ = 5 | $\epsilon$ = 2 | $\epsilon$ = 1 | $\epsilon$ = 0.5 | $\epsilon$ = 0.2 |
|------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| DP-MEPF ($\phi_1, \phi_2$) | 27.3 ± 1.5 | 26.6 ± 2.2 | 27.6 ± 2.4 | 27.6 ± 0.3 | 38.6 ± 1.9 | 64.4 ± 5.6 | 325.0 ± 15.9 |
| DP-MEPF ($\phi_1$) | 25.8 ± 2.3 | 27.1 ± 1.0 | 27.7 ± 2.2 | 28.7 ± 1.1 | 39.0 ± 0.5 | 78.4 ± 8.1 | 469.3 ± 8.8 |
| DP-MERF | 127.4 ± 1.8 | 124.4 ± 2.3 | 124.0 ± 0.8 | 126.5 ± 2.8 | 122.7 ± 1.1 | 412.8 ± 0.8 |

Table 15: Test accuracies (higher better) of ResNet9 trained on CIFAR-10 synthetic data with varying privacy guarantees. When trained on real data, test accuracy is 88.3%

| $\epsilon$ = $\infty$ | $\epsilon$ = 10 | $\epsilon$ = 5 | $\epsilon$ = 2 | $\epsilon$ = 1 | $\epsilon$ = 0.5 | $\epsilon$ = 0.2 |
|------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| DP-MEPF ($\phi_1, \phi_2$) | 47.2 ± 1.3 | 48.9 ± 3.5 | 47.9 ± 1.4 | 38.7 ± 2.3 | 28.9 ± 1.5 | 19.7 ± 3.6 | 12.4 ± 1.9 |
| DP-MEPF ($\phi_1$) | 50.8 ± 1.7 | 51.0 ± 2.1 | 48.5 ± 2.6 | 42.5 ± 0.8 | 29.4 ± 2.9 | 19.4 ± 2.9 | 13.8 ± 1.9 |
| DP-MERF | 13.2 ± 0.4 | 13.4 ± 0.4 | 13.5 ± 0.5 | 13.8 ± 1.4 | 13.1 ± 0.7 | 10.4 ± 0.5 |

Table 16: FID scores for synthetic unlabelled CIFAR-10 data

| $\epsilon$ = $\infty$ | $\epsilon$ = 10 | $\epsilon$ = 5 | $\epsilon$ = 2 | $\epsilon$ = 1 | $\epsilon$ = 0.5 | $\epsilon$ = 0.2 |
|------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| DP-MEPF ($\phi_1, \phi_2$) | 24.3 ± 1.2 | 27.1 ± 2.7 | 24.9 ± 1.0 | 26.0 ± 0.7 | 27.2 ± 3.0 | 34.8 ± 2.5 | 56.6 ± 7.9 |
| DP-MEPF ($\phi_1$) | 26.6 ± 1.6 | 26.8 ± 1.6 | 25.9 ± 0.9 | 28.9 ± 2.8 | 32.0 ± 1.7 | 38.6 ± 4.7 | 53.9 ± 2.1 |

Table 17: Unlabeled Cifar10 FID scores achieved with different feature extractors. VGG models yield the best results both in non-DP and high DP settings.

| Encoder model | #features | $\epsilon$ = $\infty$ | $\epsilon$ = 0.2 |
|----------------|-----------|-----------------|-----------------|
| VGG19 | 303104 | 24.7 | 56.2 |
| VGG16 | 276480 | 25.4 | 56.5 |
| VGG13 | 249856 | 24.4 | 58.0 |
| VGG11 | 151552 | 25.0 | 48.9 |
| ResNet50 | 199604 | 58.8 | 91.2 |
| ResNet18 | 47104 | 71.9 | 83.8 |

Table 18: FID scores achieved for CelebA 32 × 32 using a ResNet encoder with different public training sets

| ImageNet | Cifar10 | SVHN |
|----------|---------|-------|
| FID | 37 | 135 | 172 |
F Additional Plots

Below we show samples from our generated MNIST and FashionMNIST data in Figure 7 and Figure 8 respectively.

Figure 7: MNIST samples produced with DP-MEPF ($\phi_1, \phi_2$) at various levels of privacy

Figure 8: Fashion-MNIST samples produced with DP-MEPF ($\phi_1, \phi_2$) at various levels of privacy