Differentially Private Neural Tangent Kernels (DP-NTK) for Privacy-Preserving Data Generation

Yilin Yang   YANGYL17@CS.UBC.CA
Department of Computer Science,
University of British Columbia, Vancouver, Canada.

Kamil Adamczewski   KAMIL.M.ADAMCZEWSKI@GMAIL.COM
ETH Zurich, Switzerland.

Xiaoxiao Li   XIAOXIAO.LI@ECE.UBC.CA
Department of Electrical and Computer Engineering,
University of British Columbia, Vancouver, Canada.
Vector Institute, Toronto, Canada.

Danica J. Sutherland   DSUTH@CS.UBC.CA
Department of Computer Science,
University of British Columbia, Vancouver, Canada.
Alberta Machine Intelligence Institute, Edmonton, Canada.

Mijung Park   MIJUNGP@CS.UBC.CA
Department of Computer Science,
University of British Columbia, Vancouver, Canada.
Alberta Machine Intelligence Institute, Edmonton, Canada.
Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Denmark.

Abstract

Maximum mean discrepancy (MMD) is a particularly useful distance metric for differentially private data generation: when used with finite-dimensional features, it allows us to summarize and privatize the data distribution once, which we can repeatedly use during generator training without further privacy loss. An important question in this framework is, then, what features are useful to distinguish between real and synthetic data distributions, and whether those enable us to generate quality synthetic data. This work considers using the features of *neural tangent kernels (NTKs)*, more precisely *empirical* NTKs (e-NTKs). We find that, perhaps surprisingly, the expressiveness of the untrained e-NTK features is comparable to that of the features taken from pre-trained perceptual features using public data. As a result, our method improves the privacy-accuracy trade-off compared to other state-of-the-art methods, without relying on any public data, as demonstrated on several tabular and image benchmark datasets.

1. Introduction

The initial work on differentially private data generation mainly focuses on a theoretical understanding of the utility of generated data, under strong constraints on the data type and its intended use (Xiao et al., 2010; Mohammed et al., 2011; Hardt et al., 2012; Zhu et al., 2017). However, recent advances in deep generative modelling provide an appealing, more general path towards private machine learning: if trained on synthetic data samples

from a sufficiently faithful, differentially private generative model, any learning algorithm becomes automatically private. One attempt towards this goal is to privatize standard deep generative models by adding appropriate noise to each step of their training process with DP-SGD (Abadi et al., 2016). The majority of the work under this category is based on the generative adversarial networks (GAN) framework. In these approaches, the gradients of the GAN discriminator are privatized with DP-SGD, which limits the size of the discriminator for a good privacy-accuracy trade-off of DP-SGD (Xie et al., 2018; Torkzadehmahani et al., 2019; Yoon et al., 2019; Chen et al., 2020).

Another line of work builds off kernel mean embeddings (Muandet et al., 2017; Berlinet & Thomas-Agnan, 2004, Chapter 4), which summarize a data distribution by taking the mean of its feature embedding into a reproducing kernel Hilbert space. The distance between these mean embeddings, known as the Maximum Mean Discrepancy or MMD (Gretton et al., 2012), gives a distance between distributions; one way to train a generative model is to directly minimize (an estimate of) this distance (as in Dziugaite et al., 2015; Li et al., 2015).

By choosing a different kernel, the MMD "looks at" data distributions in different ways; for instance, the MMD with a short-lengthscale Gaussian kernel will focus on finely localized behaviour and mostly ignore long-range interactions, while the MMD with a linear kernel is based only on the difference between distributions' means. Choosing an appropriate kernel, then, has a strong influence on the final behaviour of a generative model. In non-private models, there has been much exploration of adversarial optimization such as GANs (Goodfellow et al., 2014, and thousands of follow-ups), which corresponds to learning a kernel to use for the MMD (Li et al., 2017; Bińkowski et al., 2018; Arbel et al., 2018).

In private settings, however, fixing the kernel for a mean embedding has a compelling advantage: if we pick a fixed kernel with a finite-dimensional feature embedding, we can privatize the kernel mean embedding of a target dataset *once*, then repeatedly use the privatized embedding during generator training without incurring any further loss of privacy due to post-processing property. This insight was first used in generative modelling in DP-MERF (Harder et al., 2021), which uses random Fourier features (Rahimi & Recht, 2007) for Gaussian kernels; Hermite polynomial features can provide a better trade-off in DP-HP (Vinaroz et al., 2022). Kernels based on perceptual features extracted from a deep network can also give far better results, if a related but public dataset is available to train that network, via DP-MEPF (Harder et al., 2023).

When no such public data is available, we would still like to use better kernels than Gaussians, which tend to have poor performance on complex datasets like natural images. In this work, we turn to a powerful class of kernels known as Neural Tangent Kernels (NTKs; Jacot et al., 2018; Lee et al., 2019; Arora et al., 2019; Chizat et al., 2019). The "empirical NTK" (e-NTK) describes the local training behaviour of a particular neural network, giving a first-order understanding of "how the network sees data" (see, e.g., Ren et al., 2022, Proposition 1, or Chizat et al., 2019). As the width of the network grows (with appropriate parameterization), this first-order behaviour dominates and the e-NTK converges to a function that depends on the architecture but not the particular weights, often called "the NTK." (Section 2 has more details.) Standard results show that in this infinite limit, training deep networks with stochastic gradient descent corresponds to kernel regression with the NTK. More relevantly for our purposes, common variants of GANs with

infinitely-wide discriminator networks reduce exactly to minimizing the MMD based on an NTK (Franceschi et al., 2022), providing significant motivation for our choice of kernel.

In addition to their theoretical motivations, NTKs have proved to be powerful general-purpose kernels for small-data classification tasks (Arora et al., 2020), perhaps because typical neural network architectures provide "good-enough" prior biases towards useful types of functions for such problems. They have similarly been shown to work well for the problem of statistically testing whether the MMD between two distributions is nonzero (and hence the distributions are different) based on samples (Jia et al., 2021; Cheng & Xie, 2021), where it can be competitive with learning a problem-specific deep kernel (Sutherland et al., 2017; Liu et al., 2020). The task of generative modelling is intimately related to this latter problem since we are aiming for a model indistinguishable from the target data (see further discussion by, e.g., Arbel et al., 2018).

In this work, we use mean embeddings based on e-NTKs as targets for our private generative model; we use e-NTKs rather than infinite NTKs for their finite-dimensional embeddings and computational efficiency. We show how to use these kernels within a private generative modelling framework, in (to the best of our knowledge) the first practical usage of NTK methods in privacy settings. Doing so yields a high-quality private generative model for several domains, outperforming models like DP-MERF and DP-HP as well as models based on DP-SGD.

2. Background

We begin by providing a brief background on differential privacy, MMD, and NTKs.

2.1 Differential Privacy (DP)

A differentially private mechanism guarantees a limit on how much information can be revealed about any single individual's participation in the dataset: formally, a mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private if for all datasets $\mathcal{D}, \mathcal{D}'$ differing by a single entry and all sets of possible outputs S for the mechanism, $\Pr[\mathcal{M}(\mathcal{D}) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta$. The level of information leak is quantified by a chosen value of ϵ and δ; smaller values correspond to stronger guarantees.

In this work, we use the *Gaussian mechanism* to ensure the output of our algorithm is DP. Given a function $h(\mathcal{D})$ with outputs in $\mathbb{R}^d$, we will enforce privacy by adding a level of noise based on the *global sensitivity* Δ_h (Dwork et al., 2006). This is defined as the maximum L_2 norm by which h can differ for neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$, $\sup \|h(\mathcal{D}) - h(\mathcal{D}')\|_2$. The mechanism's output is denoted by $\widetilde{h}(\mathcal{D}) = h(\mathcal{D}) + n$, where $n \sim \mathcal{N}(0, \sigma^2 \Delta_h^2 \mathbf{I}_d)$. The perturbed function $\widetilde{h}(\mathcal{D})$ is (ϵ, δ)-DP, where σ is a function of ϵ and δ. While $\sigma = \sqrt{2 \log(1.25/\delta)}/\epsilon$ is sufficient for $\epsilon \leq 1$, numerically computing σ given ϵ and δ using the `auto-dp` package of Wang et al. (2019) gives a smaller σ for the same privacy guarantee.

2.2 Maximum Mean Discrepancy (MMD)

For any positive definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a Hilbert space $\mathcal{H}$ and a *feature map* $\phi : \mathcal{X} \to \mathcal{H}$ such that $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$ for all $x, y \in \mathcal{X}$: the kernel k is the

inner product between feature maps (Aronszajn, 1950). We can also "lift" this definition of a feature map to distributions, defining the *(kernel) mean embedding* of P as

$$\mu_P = \mathbb{E}_{x \sim P}[\phi(x)] \in \mathcal{H}; \tag{1}$$

it exists as long as $\mathbb{E}_{x \sim P} \sqrt{k(x,x)} < \infty$ (Smola et al., 2007). One definition of the MMD is (Gretton et al., 2012)

$$\mathrm{MMD}(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}}. \tag{2}$$

If k is *characteristic* (Sriperumbudur et al., 2011), then $P \mapsto \boldsymbol{\mu}_P$ is injective, and so the MMD is a distance metric on probability measures. If k is not characteristic, it still satisfies all the requirements of a distance except that we might have $\mathrm{MMD}(P, Q) = 0$ for some $P \neq Q$.

Typically, we do not observe distributions P and Q directly, but rather estimate (2) based on i.i.d. samples $\{x_1, \ldots, x_m\} \sim P$ and $\{x_1', \ldots, x_n'\} \sim Q$. Although other estimators exist, it is often convenient to use a simple "plug-in estimator" for the mean embeddings:

$$\hat{\mu}_P = \frac{1}{m} \sum_{i=1}^{m} \phi(x_i); \quad \widehat{\mathrm{MMD}}(P, Q) = \|\hat{\mu}_P - \hat{\mu}_Q\|_{\mathcal{H}}. \tag{3}$$

For many kernels, an explicit finite-dimensional feature map $\phi : \mathcal{X} \to \mathbb{R}^d$ is available; for a simple example, the linear kernel $k(x, y) = x^\top y$ has the feature map $\phi(x) = x$ and mean embedding $\boldsymbol{\mu}_P = \mathbb{E}_{x \sim P}[x]$. The MMD then becomes simply the Euclidean distance between mean embeddings in $\mathbb{R}^d$. This setting is particularly amenable to differential privacy: we can apply the Gaussian mechanism to the estimate of the relevant mean embedding in $\mathbb{R}^d$.

The commonly-used Gaussian kernel has an infinite-dimensional embedding; DP-MERF (Harder et al., 2021) therefore approximates it with a kernel corresponding to random Fourier features that are finite-dimensional (Rahimi & Recht, 2007), while DP-HP (Vinaroz et al., 2022) uses a different approximation based on truncating Hermite polynomial features (Mehler, 1866). DP-MEPF (Harder et al., 2023) chooses a different kernel, one which is defined in the first place from finite-dimensional features extracted from a deep network. Our e-NTK features are also finite-dimensional, with a particular choice of neural network defining the kernel.

2.3 Neural Tangent Kernel (NTK)

The NTK was defined by Jacot et al. (2018), emerging from a line of previous work on optimization of neural networks. The empirical NTK, which we call e-NTK, arises from Taylor expanding the predictions of a deep network f_θ, with parameters $\theta \in \mathbb{R}^d$, and is given by

$$\mathrm{e\text{-}NTK}(x, x') = [\nabla_\theta f_\theta(x)]^\top [\nabla_\theta f_\theta(x')].$$

The gradient here is with respect to all the parameters in the network, treated as a vector in $\mathbb{R}^d$. The e-NTK is a kernel function, with an explicit embedding $\phi(x) = \nabla_\theta f_\theta(x)$.

For networks whose output has multiple dimensions (e.g. a multi-class classifier), we use f_θ above to refer to the *sum* of those outputs; this "sum-of-logits" scheme approximates the matrix-valued e-NTK (Mohamadi et al., 2022a), up to a constant scaling factor which does

not affect our usage. In PyTorch (Paszke et al., 2019), $\nabla_\theta f_\theta(x)$ can be easily computed by simply calling `autograd.grad` on the multi-output network, which sums over the output dimension if the network's output is multi-dimensional.

As appropriately-initialized networks of a fixed depth become wider, the e-NTK (a) converges from initialization to a fixed kernel, often called "the NTK", independent of θ; and (b) remains essentially constant throughout gradient descent optimization (Jacot et al., 2018; Lee et al., 2019; Arora et al., 2019; Chizat et al., 2019; Yang, 2019; Yang & Littwin, 2021). Algorithms are available to compute this limiting function exactly (Arora et al., 2019; Novak et al., 2020), but they no longer have an exact finite-dimensional embedding and can be significantly more computationally expensive to compute than e-NTKs for "reasonable-width" networks.

GANs with discriminators in an appropriate infinite limit theoretically become MMD minimizers using the NTK of the discriminator architecture (Franceschi et al., 2022). In concurrent work, Zhang et al. (2022) train generative models of a similar form directly, using the infinite NTK; unfortunately, the particular model they use – which is, although they do not quite say so, the infinite-width limit of a least-squares GAN (Mao et al., 2017) – appears much more difficult to privatize. The MMD with infinite NTKs has also proved useful in statistically testing whether the MMD between two datasets is positive, i.e. whether the distributions differ (Jia et al., 2021; Cheng & Xie, 2021).

Other applications of the e-NTK include predicting the generalization ability of a network architecture or pre-trained network for fine-tuning (Wei et al., 2022; Bachmann et al., 2022; Malladi et al., 2022), studying the evolution of networks over time (Fort et al., 2020), and in approximating selection criteria active learning (Mohamadi et al., 2022b). The infinite NTK has seen more widespread adoption in a variety of application areas; the software page of Novak et al. (2020) maintains an extensive list.

3. The DP-NTK Model

Here we present our method, which we call *differentially private neural tangent kernels (DP-NTK)* for privacy-preserving data generation.

We describe our method in the class-conditional setting; it is straightforward to translate to unconditional generation by, e.g., assigning all data points to the same class.

Following Harder et al. (2021), Vinaroz et al. (2022), Harder et al. (2023), we encode class information in the MMD objective by defining a kernel for the joint distribution over the input and label pairs in the following way. Specifically, we define

$$k((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) = k_\mathbf{x}(\mathbf{x}, \mathbf{x}') k_\mathbf{y}(\mathbf{y}, \mathbf{y}'),$$

where $k_\mathbf{y}(\mathbf{y}, \mathbf{y}') = \mathbf{y}^\top \mathbf{y}'$ for one-hot encoded labels $\mathbf{y}, \mathbf{y}'$. For the kernel on the inputs, we use the "normalized" e-NTK of a network $f_\theta : \mathcal{X} \to \mathbb{R}$:

$$k_\mathbf{x}(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}),$$
$$\text{where } \phi(\mathbf{x}) = \nabla_\theta f_\theta(\mathbf{x}) / \left\| \nabla_\theta f_\theta(\mathbf{x}) \right\|.$$

The normalization is necessary to bound the sensitivity of the mean embedding. Recall that if a network outputs vectors, e.g. logits for a multiclass classification problem, we use f_θ to denote the sum of those outputs (Mohamadi et al., 2022a).

Algorithm 1 DP-NTK

$model \leftarrow$ Generative model
$model\_NTK \leftarrow$ NTK model, randomly initialized
$Dataloader \leftarrow$ Dataloader for training set
$d \leftarrow$ Length of flattened NTK features, i.e. the size of NTK model parameters
$n \leftarrow$ Size of training data obtained by count
$n_c \leftarrow$ Class count
$batch\_size \leftarrow$ Size of batch
$n\_iter \leftarrow$ Number of iterations
mean_emb1 $\leftarrow d \times n_c$ mean embedding of training data (sensitive), initialize at 0
for Data, Label in Dataloader **do**
  for each x in Data **do**
    $\phi(x) \leftarrow$ flattened NTK features for datapoint x calculated using model_NTK calculated using `autograd.grad`
    $\phi(x) \leftarrow \frac{\phi(x)}{||\phi(x)||}$
    mean_emb1$[:, \text{Label}[x]] \mathrel{+}= \phi(x)$
  end for each
end for
noisy_mean_emb1 $\leftarrow$ mean embedding of training data with added noise via the Gaussian mechanism
for i in n_iter **do**
  $gen\_code, gen\_label \leftarrow$ code_function
  $gen\_data \leftarrow$ model(gen_code)
  mean_emb2 $\leftarrow d \times n_c$ mean embedding of generated data (non-sensitive)
  for each x in gen_data **do**
    $\phi(x) \leftarrow$ flattened NTK features for datapoint x calculated using model_NTK calculated using `autograd.grad`
    $\phi(x) \leftarrow \frac{\phi(x)}{||\phi(x)||}$
    mean_emb2$[:, \text{gen\_label}[x]] \mathrel{+}= \phi(x)$
  end for each
  MMD loss $= ||\text{noisy\_mean\_emb1} - \text{mean\_emb2}||_{2,1}^2$
  backward using loss then step
  per-iter evaluation
end for
final accuracy evaluation

Given a labelled dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$, we represent the mean embedding of the data distribution as

$$\hat{\boldsymbol{\mu}}_P = \frac{1}{m} \sum_{i=1}^m \boldsymbol{\phi}(\mathbf{x}_i)\mathbf{y}_i^\top; \tag{4}$$

this is a $d \times c$ matrix, where d is the dimensionality of the NTK features and c the number of classes.[1]

Proposition 3.1. *The global sensitivity of the mean embedding* (4) *is* $\Delta_{\boldsymbol{\mu}_P} = 2/m$.

Proof. Using the definition of the global sensitivity, which bounds for $\mathcal{D}, \mathcal{D}'$ which differ in only one entry, and the fact that y is a one-hot vector and $\boldsymbol{\phi}$ is normalized, we have that

$$
\begin{aligned}
\Delta_{\hat{\mu}_p} &= \sup_{\mathcal{D}, \mathcal{D}'} \left\| \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\phi}(\mathbf{x}_i) \mathbf{y}_i^{\top} - \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{\phi}(\mathbf{x}_j') \mathbf{y}_j'^{\top} \right\|_F \\
&= \sup_{(\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')} \left\| \frac{1}{m} \boldsymbol{\phi}(\mathbf{x}) \mathbf{y}^{\top} - \frac{1}{m} \boldsymbol{\phi}(\mathbf{x}') \mathbf{y}'^{\top} \right\|_F \\
&\leq \frac{2}{m} \sup_{(\mathbf{x}, \mathbf{y})} \left\| \boldsymbol{\phi}(\mathbf{x}) \mathbf{y}^{\top} \right\|_F \\
&= \frac{2}{m} \sup_{\mathbf{x}} \| \boldsymbol{\phi}(\mathbf{x}) \| \\
&= \frac{2}{m}. \qquad \square
\end{aligned}
$$

We use the Gaussian mechanism to privatize the mean embedding of the data distribution,

$$
\begin{aligned}
\tilde{\boldsymbol{\mu}}_P &= \hat{\boldsymbol{\mu}}_P + \mathcal{N}(0, \sigma^2 \Delta_{\boldsymbol{\mu}_P}^2 \mathbf{I}_d) \\
&= \hat{\boldsymbol{\mu}}_P + \mathcal{N}\left(0, \frac{4\sigma^2}{m^2} \mathbf{I}_d\right).
\end{aligned}
$$

The privacy parameter σ is a function of ϵ and δ, which we numerically compute using the `auto-dp` package of Wang et al. (2019).

Our generator G produces an input conditioned on a generated label $\mathbf{y}_i'$, i.e. $G(\mathbf{z}_i, \mathbf{y}_i') \mapsto \mathbf{x}_i'$, where $\mathbf{y}_i'$ is drawn from a uniform distribution over the c classes and each entry of $\mathbf{z}_i \in \mathbb{R}^{d_z}$ is drawn from a standard Gaussian distribution. Similar to (4), the mean embedding of the synthetic data distribution is given by

$$
\hat{\boldsymbol{\mu}}_Q = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i') \mathbf{y}_i'^{\top}. \tag{5}
$$

Our privatized MMD loss is given by

$$
\widetilde{\mathrm{MMD}}^2(P, Q) = \left\| \tilde{\boldsymbol{\mu}}_P - \hat{\boldsymbol{\mu}}_Q \right\|_F^2, \tag{6}
$$

where F is the Frobenius norm. This loss is differentiable (as was proved in the NTK case by Franceschi et al., 2022; we compute the derivative with standard automatic differentiation

---

1. This corresponds to a feature space of $d \times c$ matrices, where we use the Frobenius norm and inner product; though this is itself a Hilbert space, this exactly agrees with "flattening" these matrices into $\mathbb{R}^{dc}$ and then operating in that Euclidean space.

systems). We minimize (6) with respect to the parameters of the generator G with standard variations on stochastic gradient descent, where $\tilde{\boldsymbol{\mu}}_P$ remains a constant vector but $\hat{\boldsymbol{\mu}}_Q$ is computed based on a new batch of generator outputs at each step.

For class-imbalanced and heterogeneous datasets, such as the tabular datasets we consider, we use the modified mean embeddings of Harder et al. (2021, Sections 4.2-4.3), replacing their random Fourier features with our e-NTK features.

4. Theoretical Analysis

Existing results on this class of models measure the quality of a generative model in terms of its squared MMD to the target distribution. How useful is that as a metric? The origin of the name "maximum mean discrepancy" is because we have in general that

$$\text{MMD}(P, Q) = \sup_{f:\|f\|_k \leq 1} \mathbb{E}_{x \sim P} f(x) - \mathbb{E}_{y \sim Q} f(y),$$

where $\|f\|_k$ gives the norm of a function $f : \mathcal{X} \to \mathbb{R}$ under the kernel k. If the squared MMD is small, no function with a small norm under the kernel k can strongly distinguish P and Q. But when our kernel is the NTK of a wide neural network, the set of functions with small kernel norm is exactly the functions which can be learned by gradient descent in bounded time (also see more detailed discussion by Cheng & Xie, 2021). Thus, if P is similar to Q, (roughly) no neural network of the architecture used by our discriminator can distinguish the two distributions. Although our e-NTK kernel is finite-dimensional and hence not characteristic, this implies that distributions with small MMD under the NTK should model the distribution well for practical usage.

We can bound this MMD between the target distribution and the private generative model in two steps: the gap between the private and the non-private model, plus the gap between the non-private model and the truth.

For the extra loss induced by privacy, we can use the theoretical analysis of Harder et al. (2023), which is general enough to also apply to DP-NTK. Their main result shows the following in our setting.

Proposition 4.1. *Fix a target distribution P, and let $\mathcal{Q}$ denote some class of probability distributions, e.g. all distributions which can be realized by setting the parameters of a generator network.*

Let $\tilde{Q} \in \arg\min_{Q \in \mathcal{Q}} \widetilde{\text{MMD}}(P, Q)$ minimize the private loss (6), and let $\hat{R} \in \arg\min_{Q \in \mathcal{Q}} \widehat{\text{MMD}}(P, Q)$ minimize the non-private loss (3).

Then the sub-optimality of $\tilde{Q}$ relative to $\hat{R}$ is given by[2]

$$\widehat{\text{MMD}}^2(P, \tilde{Q}) - \widehat{\text{MMD}}^2(P, \hat{R}) = \mathcal{O}_p\left(\frac{\sigma^2 d}{m^2} + \frac{\sigma \sqrt{d}}{m} \widehat{\text{MMD}}(P, \hat{R})\right). \tag{7}$$

The notation $A = \mathcal{O}_p(B)$ means roughly that with any constant probability, $A = \mathcal{O}(B)$. Note that the MMD in the latter term is bounded by $\sqrt{2}$, so we can always achieve a

---

2. See their Proposition A.5 for specific constants and plug in $\text{Tr}(\Sigma) = 4d\sigma^2/m^2$, $\|\Sigma\|_F = 4\sqrt{d}\sigma^2/m^2$, $\|\Sigma\|_{op} = 4\sigma^2/m^2$.

$\mathcal{O}_p(1/m)$ rate for the squared MMD. If $\text{MMD}(P, \hat{R}) = 0$, though – the "interpolating" case – this "optimistic" rate shows that the private model is only $\mathcal{O}_p(1/m^2)$ worse than the non-private one, as measured on the particular sample.

Thus, we know that the private minimizer will approximately minimize the non-private loss. The sub-optimality in that minimization decays at a rate much faster than the known rates for the convergence of the non-private minimizer to the best possible model. The rates obtained by e.g. Dziugaite et al. (2015) have complex dependence on the generator architecture, but never give a rate for $\text{MMD}^2(P, \hat{R}) - \inf_{Q \in \mathcal{Q}} \text{MMD}^2(P, Q)$ faster than $\mathcal{O}_p(1/\sqrt{m})$; this is far slower than our $\mathcal{O}_p(1/m)$ or even $\mathcal{O}_p(1/m^2)$ rate for the extra loss due to privacy, so the private model is (asymptotically) not meaningfully worse than the non-private one.

5. Experiments

We test our method, DP-NTK, on popular benchmark image datasets such as MNIST, FashionMNIST, CelebA, and CIFAR10, as well as on 8 benchmark tabular datasets. For MNIST/FMNIST we use a Conditional CNN as the generator. We use ResNet18 as our generator for Cifar10 and CelebA, and a fully connected network for tabular data: for homogenous data, we use 2 hidden layers + ReLU+ batch norm, or 3 hidden layers + ReLU + batch norm plus an additional sigmoid layer for the categorical features for heterogeneous data.

Our code is at `https://github.com/FreddieNeverLeft/DP-NTK`. The readme file of this repository and the section Appendix A contain hyper-parameter settings (e.g., the architectural choices used for the model whose e-NTK we take) for reproducibility.

5.1 Generating MNIST and FashionMNIST Images

For our MNIST and FashionMNIST experiments, we choose the e-NTK of a fully-connected 1-hidden-layer neural network architecture with a width parameter w, for simplicity. As the NTK features considered here are uniquely related to the width of the NTK network, for MNIST and F-MNIST data we conduct two groups of experiments, one where we test the accuracy of the model under the same Gaussian mechanism noise but with different NTK widths w, and the other with the same width ($w = 800$) but with varying privacy noise levels. In this setting, we will also compare results with other DP image-generating methods that do not rely on public data for pre-training, as well as DP-MEPF which uses public data to train a feature extractor model.

Following prior work such as Harder et al. (2021), Vinaroz et al. (2022), Harder et al. (2023), we quantitatively measure the quality of samples by training a classifier on the synthetic data and then measuring its accuracy when applied to real data. Here, we use two image classifiers: a multilayer perceptron (MLP) and logistic regression.

Privacy-Width Trade-off   As discussed e.g. by Vinaroz et al. (2022), applying the Gaussian mechanism in feature spaces of higher dimension causes our model to become less accurate. This necessitates a trade-off in the feature dimensionality: large dimensions can lead to overwhelming amounts of added noise, while small dimensions may be inadequate to serve as a loss for image generation.

MNIST				
	$w = 100$	$w = 400$	$w = 800$	$w = 1000$
Log Reg	0.8423	0.8404	0.84	0.8387
MLP	0.8824	0.8811	0.88	0.8805
FashionMNIST				
	$w = 100$	$w = 400$	$w = 800$	$w = 1000$
Log Reg	0.7640	0.7642	0.7663	0.7643
MLP	0.7773	0.7806	0.7838	0.7768

Table 1: Accuracy under different widths of the NTK 1 layer NN, with $(10, 10^{-5})$ DP

Figure 1: Generated samples of MNIST and FashionMNIST from DP-NTK with different widths w; all samples use the same DP noise level ($\epsilon = 10$, $\delta = 10^{-5}$).
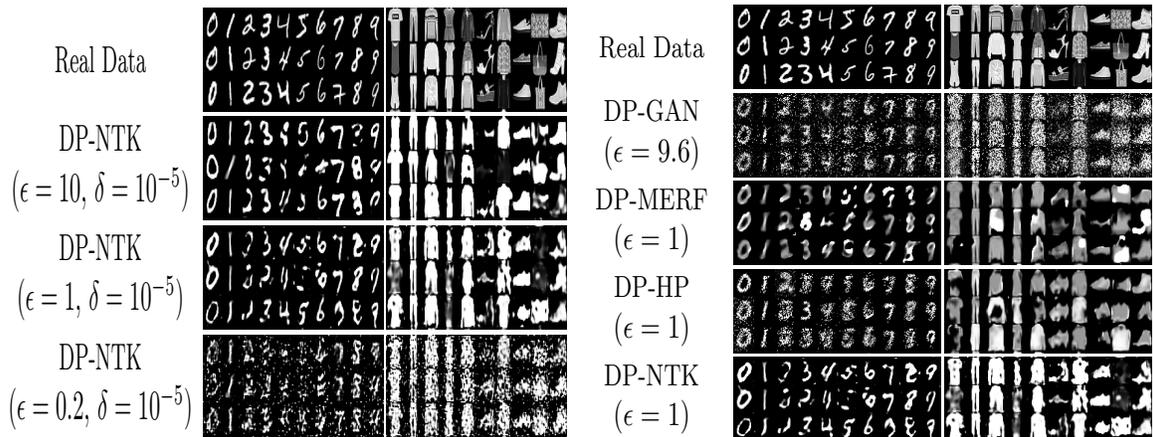
(a) Generated samples for MNIST and FashionMNIST from DP-NTK, with the same width ($w =$ 800) and different DP levels.

(b) Generated samples for MNIST and FashionMNIST from DP-NTK and comparison models

Figure 2: DP-NTK under different DP levels (left) and comparison results with different models (right) for MNIST and FashionMNIST

Method	DP-ϵ	MNIST		Fashion-MNIST	
		Log Reg	MLP	Log Reg	MLP
DP-NTK (ours)	0.2	*0.804*	*0.794*	0.634	0.694
DPDM (Dockhorn et al., 2023)	0.2	**0.81**	**0.817**	0.704	**0.713**
PEARL (Liew et al., 2021)	0.2	0.762	0.771	0.700	0.708
DP-MERF (Harder et al., 2021)	0.2	0.772	0.768	**0.714**	0.696
DP-NTK (ours)	1	*0.8324*	*0.8620*	*0.7596*	*0.7645*
DPDM (Dockhorn et al., 2023)	1	**0.867**	**0.916**	**0.763**	**0.769**
PEARL (Liew et al., 2021)	1	0.76	0.796	0.744	0.740
DP-HP (Vinaroz et al., 2022)	1	0.807	0.801	0.7387	0.7215
DP-MERF (Harder et al., 2021)	1	0.769	0.807	0.728	0.738
DP-NTK (ours)	10	*0.8400*	*0.8800*	*0.7663*	*0.7838*
DPDM (Dockhorn et al., 2023)	10	**0.908**	**0.948**	**0.811**	**0.830**
PEARL (Liew et al., 2021)	10	0.765	0.783	0.726	0.732
DP-Sinkhorn (Cao et al., 2021)	10	0.828	0.827	0.751	0.746
DP-CGAN (Torkzadehmahani et al., 2019)	10	0.6	0.6	0.51	0.5
DP-HP (Vinaroz et al., 2022)	10	0.8082	0.8038	0.7387	0.7168
DP-MERF (Harder et al., 2021)	10	0.794	0.783	0.755	0.745
GS-WGAN (Chen et al., 2020)	10	0.79	0.79	0.68	0.65

Table 2: Performance comparison on MNIST and F-MNIST dataset averaged over five independent runs, with NTK width fixed at 800. Here δ is fixed at 10^{-5}.

Figure 1 and Table 1 show that for both MNIST and FashionMNIST, changing the width does somewhat affect the final accuracy and image quality, but this effect is very minimal. Therefore, for subsequent experiments, we choose a width of 800 as a good compromise.

Method	DP-ϵ	MNIST		Fashion-MNIST	
		Log Reg	MLP	Log Reg	MLP
DP-NTK (ours)	0.2	**0.804**	**0.794**	0.634	0.694
DP-MEPF (Harder et al., 2023)	0.2	0.80	0.76	**0.73**	**0.7**
DP-NTK (ours)	1	**0.8324**	0.8620	**0.7596**	**0.7645**
DP-MEPF (Harder et al., 2023)	1	0.82	**0.89**	0.75	0.75
DP-NTK (ours)	10	**0.8400**	0.8800	**0.7663**	**0.7838**
DP-MEPF (Harder et al., 2023)	10	0.83	**0.89**	0.76	0.76

Table 3: Performance comparison on MNIST and F-MNIST dataset for our method with NTK width fixed at 800 and DP-MEPF (Harder et al., 2023). Here δ is fixed at 10^{-5}.

Varying privacy levels Table 2 and Figure 2a show our model's performance under different levels of privacy. Other than for FashionMNIST with $\epsilon = 0.2$, the performance of our model does not degrade significantly as the privacy requirement becomes more stringent.

Table 2 also compares with several existing models, while Figure 2b visually compares results with DP-GAN (Xie et al., 2018), DP-MERF (Harder et al., 2021) and DP-HP (Vinaroz et al., 2022). We see that even with simple architectures, DP-NTK broadly performs better than other high-accuracy models, and generates comprehensible images. Although there is still a distance from the current SOTA (Dockhorn et al., 2023), we show that DP-NTK still offers competitive performance under different privacy levels on MNIST and FashionMNIST. We also compare the performance of DP-MEPF (that relies on public data for pre-training a feature extractor model) and DP-NTK on MNIST and FashionMNIST datasets in Table 3.
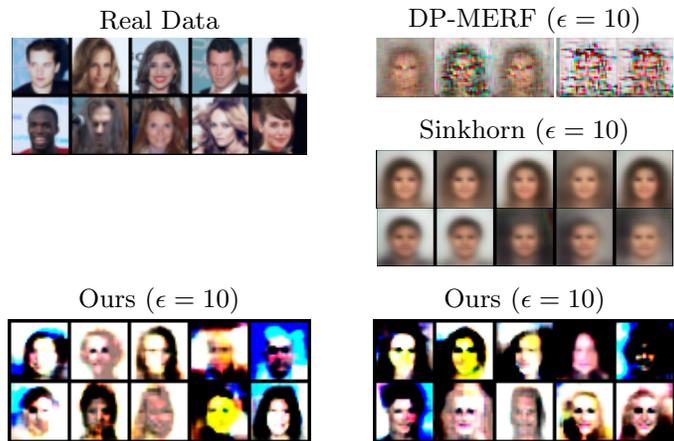

Figure 3: Synthetic 32×32 CelebA samples generated at different levels of privacy. Samples for DP-MERF and DP-Sinkhorn are taken from Cao et al. (2021). Our method yields samples of higher visual quality than the comparison methods. The FID for the proposed method is 75. FID for DP-Sinkhorn is 189. FID for DP-MERF is 274.
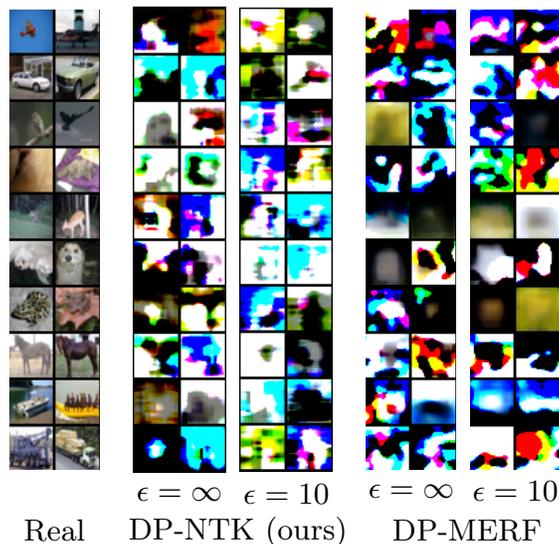
	$\epsilon = \infty$	$\epsilon = 10$	$\epsilon = \infty$	$\epsilon = 10$
Real	DP-NTK (ours)		DP-MERF	

Figure 4: The generated and real images for the CIFAR-10 dataset. The FID scores for the proposed method are 104 ($\epsilon = \infty$) and 107 ($\epsilon = 10$), respectively. For DP-MERF, they are 127 ($\epsilon = \infty$) and 141 ($\epsilon = 10$).

5.2 Generating CelebA and CIFAR10 Images

Image datasets beyond MNIST are notoriously challenging for differentially private data generation. In the Appendix, Figure 3 on CelebA (at 32×32 resolution). On CelebA, DP-MERF and DP-Sinkhorn give blurry, near-uniform samples, with DP-MERF also containing significant amounts of very obvious pixel-level noise. Our method, by contrast, generates sharper, more diverse images with some distinguishable facial features, although generation quality remains far behind that of non-private models.

Figure 4 shows results on CIFAR-10, where the difficulty lies in having relatively few samples for a wide range of image objects. Our samples, while again far from the quality of non-private generators, are less blurry than DP-MERF and contain shapes reminiscent of real-world objects.

5.3 Generating Tabular Data

For DP-NTK, we explore the performance of the algorithm on eight different imbalanced tabular datasets with both numerical and categorical input features. The numerical features on those tabular datasets can be either discrete (e.g. age in years) or continuous (e.g. height), and the categorical ones may be binary (e.g. drug vs placebo group) or multi-class (e.g. nationality).

For evaluation, we use ROC (area under the receiver characteristic curve) and PRC (area under the precision-recall curve) for datasets with binary labels, and F1 score for datasets with multi-class labels. Table 4 shows the average over the 12 classifiers trained on the generated samples (also averaged over 5 independent seeds). For most of the datasets, DP-NTK outperforms the benchmark methods.

	Real		DP-CGAN $(1, 10^{-5})$-DP		DP-GAN $(1, 10^{-5})$-DP		DP-MERF $(1, 10^{-5})$-DP		DP-HP $(1, 10^{-5})$-DP		**DP-NTK** $(1, 10^{-5})$-DP	
adult	0.786	0.683	0.509	0.444	0.511	0.445	0.642	0.524	0.688	**0.632**	**0.695**	0.557
census	0.776	0.433	0.655	0.216	0.529	0.166	0.685	0.236	0.699	0.328	**0.71**	**0.424**
cervical	0.959	0.858	0.519	0.200	0.485	0.183	0.531	0.176	0.616	0.31	**0.631**	**0.335**
credit	0.924	0.864	0.664	0.356	0.435	0.150	0.751	0.622	0.786	0.744	**0.821**	**0.759**
epileptic	0.808	0.636	0.578	0.241	0.505	0.196	0.605	0.316	0.609	**0.554**	**0.648**	0.326
isolet	0.895	0.741	0.511	0.198	0.540	0.205	0.557	0.228	**0.572**	**0.498**	0.53	0.205
	F1		F1		F1		F1		F1		F1	
covtype	0.820		0.285		0.492		0.467		0.537		**0.552**	
intrusion	0.971		0.302		0.251		**0.892**		0.890		0.717	

Table 4: Performance comparison on Tabular datasets averaged over five independent runs. The top six datasets contain binary labels while the bottom two datasets contain multi-class labels. The metric for the binary datasets is ROC and PRC and for the multiclass datasets F1 score.

6. Summary and Discussion

We introduced DP-NTK, a new model for differentially private generative modelling. NTK features are an excellent fit for this problem, providing a data-independent representation that performs well on a variety of tasks: DP-NTK greatly outperforms existing private generative models (other than those relying on the existence of similar public data for pre-training) for relatively simple image datasets and a majority of tabular datasets that we evaluated.

There are, however, certainly avenues that could improve the DP-NTK model in future work. One path might be finding a way to use the infinite NTK, which tends to have better performance on practical problems than e-NTKs at initialization (Arora et al., 2020).

When related public datasets are available for pre-training, it would also be helpful for DP-NTK to be able to incorporate that information, as Harder et al. (2023) have argued is important. Indeed, by exploiting this information, their model outperforms ours on some image datasets. One strategy would be to take the e-NTK of a pre-trained model on that related dataset; our initial attempt at this did not help much, however, and we leave further investigation to future work.

Acknowledgement

Appendix A. Hyperparameters Used in Experiments

dataset	iter	d_code	NTK width	batch	lr	eps	architecture
dmnist	2000	5	800	5000	0.01	10, 1, 0.2	fc_1l
fmnist	2000	5	800	5000	0.01	10, 1, 0.2	fc_1l
celeba	20000	141	3000_200	1000	0.01	10	fc_2l
cifar10	40000	201	3000_200	1000	0.01	10 1	fc_2l
cifar10	20000	31	800_1000	200	0.01	None	fc_2l
adult	50	11	30_200	200	0.01	1	cnn_2l
census	2000	21	30_20	200	0.01	1	cnn_2l
cervical	500	11	800_1000	200	0.01	1	cnn_2l
credit	500	11	1500	200	0.01	1	fc_1l
epileptic	2000	101	50_20	200	0.01	1	cnn_2l
isolet	1000	21	10_20	200	0.01	1	cnn_2l
covtype	1000	101	100_20	200	0.01	1	cnn_2l
intrusion	1000	21	30_1000	200	0.01	1	fc_2l

Table 5: Best Hyperparamters used at different seeds for our experiments, see our repo for details.

References

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

Arbel, M., Sutherland, D. J., Bińkowski, M., & Gretton, A. (2018). On gradient regularizers for MMD GANs. In *NeurIPS*.

Aronszajn, N. (1950). Theory of reproducing kernels. *Trans Am Math Soc*, *68*(3), 337–404.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., & Wang, R. (2019). On exact computation with an infinitely wide neural net. In *NeurIPS*.

Arora, S., Du, S. S., Li, Z., Salakhutdinov, R., Wang, R., & Yu, D. (2020). Harnessing the power of infinitely wide deep nets on small-data tasks. In *ICLR*.

Bachmann, G., Hofmann, T., & Lucchi, A. (2022). Generalization through the lens of leave-one-out error. In *ICLR*.

Berlinet, A., & Thomas-Agnan, C. (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer New York.

Bińkowski, M., Sutherland, D. J., Arbel, M., & Gretton, A. (2018). Demystifying MMD GANs. In *ICLR*.

Cao, T., Bie, A., Vahdat, A., Fidler, S., & Kreis, K. (2021). Don't generate me: Training differentially private generative models with Sinkhorn divergence. In *NeurIPS*.

Chen, D., Orekondy, T., & Fritz, M. (2020). GS-WGAN: A gradient-sanitized approach for learning differentially private generators. In *NeurIPS*.

Cheng, X., & Xie, Y. (2021). Neural tangent kernel maximum mean discrepancy. In *NeurIPS*.

Chizat, L., Oyallon, E., & Bach, F. (2019). On lazy training in differentiable programming. In *NeurIPS*.

Dockhorn, T., Cao, T., Vahdat, A., & Kreis, K. (2023). Differentially private diffusion models. *Transactions on Machine Learning Research*.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., & Naor, M. (2006). Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*.

Dziugaite, G. K., Roy, D. M., & Ghahramani, Z. (2015). Training generative neural networks via maximum mean discrepancy optimization. In *UAI*.

Fort, S., Dziugaite, G. K., Paul, M., Kharaghani, S., Roy, D. M., & Ganguli, S. (2020). Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *NeurIPS*.

Franceschi, J.-Y., De Bézenac, E., Ayed, I., Chen, M., Lamprier, S., & Gallinari, P. (2022). A neural tangent kernel perspective of GANs. In *ICML*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., & Bengio, Y. (2014). Generative adversarial nets. In *NeurIPS*.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *JMLR*, *13*(25), 723–773.

Harder, F., Adamczewski, K., & Park, M. (2021). DP-MERF: Differentially Private Mean Embeddings with Random Features for Practical Privacy-preserving Data Generation. In *AISTATS*.

Harder, F., Jalali, M., Sutherland, D. J., & Park, M. (2023). Pre-trained perceptual features improve differentially private image generation. *Transactions on Machine Learning Research*.

Hardt, M., Ligett, K., & Mcsherry, F. (2012). A simple and practical algorithm for differentially private data release. In *NeurIPS*.

Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*.

Jia, S., Nezhadarya, E., Wu, Y., & Ba, J. (2021). Efficient statistical tests: A neural tangent kernel approach. In *ICML*.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., & Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. In *NeurIPS*.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., & Poczos, B. (2017). MMD GAN: Towards Deeper Understanding of Moment Matching Network. In *NeurIPS*.

Li, Y., Swersky, K., & Zemel, R. (2015). Generative moment matching networks. In *ICML*.

Liew, S. P., Takahashi, T., & Ueno, M. (2021). PEARL: Data Synthesis via Private Embeddings and Adversarial Reconstruction Learning. In *ICLR*.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A., & Sutherland, D. J. (2020). Learning deep kernels for non-parametric two-sample tests. In *ICML*.

Malladi, S., Wettig, A., Yu, D., Chen, D., & Arora, S. (2022). A kernel-based view of language model fine-tuning.. arXiv:2210.05643.

Mao, X., Li, Q., & and, H. X. (2017). Least squares generative adversarial networks. In *ICCV*.

Mehler, F. G. (1866). Ueber die entwicklung einer function von beliebig vielen variablen nach laplaceschen functionen höherer ordnung. *Journal für die reine und angewandte Mathematik*.

Mohamadi, M. A., Bae, W., & Sutherland, D. J. (2022a). A fast, well-founded approximation to the empirical neural tangent kernel.. arXiv:2206.12543.

Mohamadi, M. A., Bae, W., & Sutherland, D. J. (2022b). Making look-ahead active learning strategies feasible with neural tangent kernels. In *NeurIPS*. arXiv:2206.12569.

Mohammed, N., Chen, R., Fung, B. C., & Yu, P. S. (2011). Differentially private data release for data mining. In *KDD*, New York, NY, USA.

Muandet, K., Fukumizu, K., Sriperumbudur, B., & Schölkopf, B. (2017). Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning, 10*(1-2), 1–141. arXiv:1605.09522.

Novak, R., Xiao, L., Hron, J., Lee, J., Alemi, A. A., Sohl-Dickstein, J., & Schoenholz, S. S. (2020). Neural tangents: Fast and easy infinite neural networks in python. In *ICLR*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Rahimi, A., & Recht, B. (2007). Random features for large-scale kernel machines. In *NeurIPS*.

Ren, Y., Guo, S., & Sutherland, D. J. (2022). Better supervisory signals by observing learning paths. In *ICLR*. arXiv:2203.02485.

Smola, A., Gretton, A., Song, L., & Scholkopf, B. (2007). A Hilbert space embedding for distributions. In *ALT*.

Sriperumbudur, B. K., Fukumizu, K., & Lanckriet, G. R. (2011). Universality, characteristic kernels and rkhs embedding of measures.. *Journal of Machine Learning Research, 12*(7).

Sutherland, D. J., Tung, H.-Y., Strathmann, H., De, S., Ramdas, A., Smola, A., & Gretton, A. (2017). Generative models and model criticism via optimized maximum mean discrepancy. In *ICLR*.

Torkzadehmahani, R., Kairouz, P., & Paten, B. (2019). Dp-cgan: Differentially private synthetic data and label generation. In *CVPR Workshops*.

Vinaroz, M., Charusaie, M.-A., Harder, F., Adamczewski, K., & Park, M. J. (2022). Hermite polynomial features for private data generation. In *ICML*.

Wang, Y.-X., Balle, B., & Kasiviswanathan, S. P. (2019). Subsampled Rényi differential privacy and analytical moments accountant. In *AISTATS*.

Wei, A., Hu, W., & Steinhardt, J. (2022). More than a toy: Random matrix models predict how real-world neural representations generalize. In *ICML*.

Xiao, Y., Xiong, L., & Yuan, C. (2010). Differentially private data release through multi-dimensional partitioning. In *Secure Data Management*, pp. 150–168. Springer Berlin Heidelberg.

Xie, L., Lin, K., Wang, S., Wang, F., & Zhou, J. (2018). Differentially private generative adversarial network.. arXiv:1802.06739.

Yang, G. (2019). Tensor Programs I: Wide feedforward or recurrent neural networks of any architecture are Gaussian processes. In *NeurIPS*.

Yang, G., & Littwin, E. (2021). Tensor Programs IIb: Architectural universality of neural tangent kernel training dynamics. In *ICML*.

Yoon, J., Jordon, J., & van der Schaar, M. (2019). PATE-GAN: Generating synthetic data with differential privacy guarantees. In *ICLR*.

Zhang, Y.-R., Su, R.-Y., Chou, S. Y., & Wu, S.-H. (2022). Single-level adversarial data synthesis based on neural tangent kernels.. arXiv:2204.04090.

Zhu, T., Li, G., Zhou, W., & Yu, P. S. (2017). Differentially private data publishing and analysis: A survey. *IEEE Transactions on Knowledge and Data Engineering, 29*(8), 1619–1638.