Non-parametric estimation of Jensen-Shannon Divergence in Generative Adversarial Network training

Mathieu Sinn
IBM Research
Dublin, Ireland

Ambrish Rawat
IBM Research
Dublin, Ireland

Abstract

Generative Adversarial Networks (GANs) have become a widely popular framework for generative modelling of high-dimensional datasets. However their training is known to be difficult. This work presents a rigorous statistical analysis of GANs providing straight-forward explanations for common training pathologies such as vanishing gradients. Furthermore, it proposes a new training objective, Kernel GANs, and demonstrates its practical effectiveness on real-world data sets. A key element in the analysis is the distinction between training with respect to the (unknown) data distribution, and its empirical counterpart. To overcome issues in GAN training, we pursue the idea of smoothing the Jensen-Shannon Divergence (JSD) by incorporating noise in the input distributions of the discriminator. As we show, this effectively leads to an empirical version of the JSD in which the true and the generator densities are replaced by kernel density estimates, which leads to Kernel GANs.

1 INTRODUCTION

Generative Adversarial Networks (GANs), introduced by Goodfellow et al. (2014a), have become a widely popular framework for generative modeling using deep neural networks. While practitioners find that GANs – particularly for image data – produce sharp and realistic samples, it is well recognized that GANs are difficult to train. Key challenges are: vanishing gradients, local optima leading to mode collapse, high sensitivity to hyperparameters, and finding the right balance between generator and discriminator training in the adversarial set-up (Dinh et al. (2016); Goodfellow (2016); Goodfellow et al. (2014b); Metz et al. (2016); Radford et al. (2015a); Salimans et al. (2016)).

Various authors have proposed practical modifications of GAN training to address these issues. However, only recently have authors begun to analyze them mathematically and develop principled solutions. An important step in this direction was the work by Arjovsky and Bottou (2017), which led to the idea of Wasserstein GANs as elaborated in Arjovsky et al. (2017) and further developed by Gulrajani et al. (2017). Two important insights were: 1) training the discriminator in GANs till optimality may provably result in vanishing gradients, and 2) the Jensen-Shannon Divergence (JSD) doesn't yield meaningful information about convergence of distributions if their intersection with the support of the limit-distribution has measure zero. Another important contribution was the work by Metz et al. (2016), who proposed to unroll discriminators in the GAN training objective in order to avoid degenerate optima and vanishing gradients.

Contributions. Our work makes three major contributions.

- First, a rigorous mathematical framework to analyze GANs, which yields a remarkably simple explanation of the vanishing gradient problem.
- Second, a novel training objective, Kernel GANs, backed with a principled theoretical analysis along with an empirical study that highlights practical aspects of Kernel GAN training.
- Finally, experimentation with different training setups that scale Kernel GANs to large datasets and establish their practical usefulness.

Our mathematical framework allows us to formulate the results in Arjovsky and Bottou (2017) more generally, and it addresses obscurities in the original the-

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

ory developed in Goodfellow et al. (2014a). It appears that a major source of confusion has been the missing distinction between the GAN training objective with respect to the (unknown) target density, and its empirical counterpart used in practical training. This was recently independently pointed out by Arora et al. (2017), however, without deriving the implications stated in the present work.

To overcome GAN training pathologies, we analyze approaches for smoothing the JSD in the training objective. We pursue the idea of adding noise terms to the inputs of the discriminator. As we show, this leads to an empirical version of the JSD in which the true and the generator densities are replaced by kernel density estimates. We establish almost sure convergence of this Kernel GAN objective and conditions under which it is asymptotically unbiased.

We support the theoretical analysis with extensive experimentation. Particular emphasis is on understanding the effect of the kernel bandwidth in the training algorithm. We also study the generalization of Kernel GANs by comparing the fidelity and the diversity of generated samples with baseline methods. Furthermore, we extend the setup to enable generative modelling of large-scale real-world datasets.

Related work. In its attempt to establish a rigorous mathematical framework for understanding properties of GANs, this work is related to Arjovsky and Bottou (2017). However, our framework is more general, e.g., many of the results (Theorem 2.1-2.5 in particular) cover both discrete and continuous distributions, and no parametric family assumptions are imposed on the generators. Moreover we clarify the consequences of working with empirical distributions in training, which leads to a remarkably simple explanation of the pathologies discussed in Arjovsky and Bottou (2017).

Similar to Wasserstein GANs (Arjovsky et al. (2017)), our approach overcomes pathologies in GAN training by smoothing the discrepancy between distributions with disjoint support. However, our training objective can be regarded as a non-parameteric estimate of JSD, which allows for the study of asymptotic properties (the first of its kind in the GAN literature as far as we know). Among all the different variants of GANs, we find our approach has the closest resemblance with the works by Dziugaite et al. (2015), Li et al. (2015) and, most recently, Li et al. (2017), which also plugs kernel density estimates into the GAN training objective. However, these authors optimize generative models with respect to the Maximum Mean Discrepancy (MMD) criterion. Contrastingly, our approach retains the adversarial setup; in fact, it can be regarded as unrolling the discriminator in the GAN training objective till optimality, which is similar in spirit to the methodology proposed by Metz et al. (2016). Our approach involves annealing strategies for controlling the complexity of kernels, which is similar in spirit to AdaGANs (Tolstikhin et al. (2017)) where the complexity of distribution is controlled by gradually adding components to a mixture model.

Outline of this paper. The mathematical framework and theoretical findings are established in Section 2, which also introduces the novel training objective. Section 3 provides practical aspects of training with respect to that objective, and discusses experimental results. Section 4 concludes the paper. Proofs, implementation details and additional experiments are included in the Appendix.

2 THEORETICAL ANALYSIS

2.1 Preliminaries

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Consider measurable spaces $(\mathcal{X}, \mathcal{A})$ (the **output space**) and $(\mathcal{Z}, \mathcal{C})$ (**latent space**). Let $\mathcal{G}$ denote a set of measurable functions $g : \mathcal{Z} \to \mathcal{X}$ (**generators**), and $\mathcal{D}$ a set of measurable functions $d : \mathcal{X} \to [0,1]$ (**discriminators**). Let μ be a measure on $(\mathcal{X}, \mathcal{A})$. For the remainder of this paper, let X and Z be fixed random variable from $(\Omega, \mathcal{F}, \mathbb{P})$ onto $(\mathcal{X}, \mathcal{A})$ and $(\mathcal{Z}, \mathcal{C})$, respectively. We will make frequent use of the following assumptions:

(A1) The distribution of X is absolutely continuous with respect to μ.

(A2) For every $g \in \mathcal{G}$, the distribution of $g(Z)$ is absolutely continuous with respect to μ.

As a consequence, X has a μ-density p and $g(Z)$ has a μ-density $p^{(g)}$ for every $g \in \mathcal{G}$. Given iid samples $X_1, X_2, \ldots, X_n$ from the distribution of X, our goal is to learn a generator g such that $\mathbb{P}(g(Z) \in A) = \mathbb{P}(X \in A)$ for all $A \in \mathcal{A}$, or, equivalently, $p = p^{(g)}$ μ-almost everywhere. The next theorem establishes the existence of such a g under the following assumptions:

(A3) $\mathcal{X}$ is a Peano space, i.e., $\mathcal{X}$ is a compact, connected, and locally connected metric space.

(A4) $\mathcal{X}$ is the support of X, i.e., there doesn't exist an $x \in \mathcal{X}$ with an open neighborhood B_x in the topology of $\mathcal{X}$ such that $\mathbb{P}(X \in B_x) = 0$.

Theorem 2.1. *Suppose that* (A1)-(A4) *hold. Moreover suppose that $\mathcal{Z} = [0,1]$, $\mathcal{C}$ is the Borel σ-algebra on $[0,1]$, and Z is uniformly distributed on $\mathcal{Z}$. Then there exists a continuous surjection $g : \mathcal{Z} \to \mathcal{X}$ such that $\mathbb{P}(g(Z) \in A) = \mathbb{P}(X \in A)$ for all $A \in \mathcal{A}$.*

There has been some confusion in the GAN literature about the exact conditions that are required to obtain this result. For example, Goodfellow (2016) states that the "the only requirements" for $p^{(g)}$ to have "full support" on $\mathcal{X}$ are - the dimension of $\mathcal{Z}$ be "at least as large as the dimension of $\mathcal{X}$", and g be differentiable. This isn't accurate as $\mathcal{Z}$ may have smaller dimension, as long as its cardinality is not smaller than the one of $\mathcal{X}$, and the distribution of Z is non-atomic. Differentiability of g is not required in theory. To obtain an invertible and differentiable mapping g, the dimension of $\mathcal{Z}$ must not be smaller than the dimension of $\mathcal{X}$. The result in Theorem 2.1 relies on a construction using space-filling curves, which aren't differentiable.

2.2 GAN Training – Theoretical Case

The GAN approach (Goodfellow et al. (2014a)) for learning g is as follows: for $d \in \mathcal{D}$ and $g \in \mathcal{G}$ let

$$V(d,g) := \mathbb{E}\big[\log(d(X))\big] + \mathbb{E}\big[\log(1-d(g(Z)))\big]. \quad (1)$$

The relation of $V(d,g)$ to density ratio estimation (which becomes apparent in equation (4) below) is discussed in Mohamed and Lakshminarayanan (2016). Intuitively, we wish the discriminator $d(x)$ to be close to 1 if x is more likely under the distribution of X, and close to 0 if x is more likely under the distribution of $g(Z)$. Hence, the optimal d given a fixed generator g would attempt to maximize $V(\cdot, g)$, and the optimal g is the one which solves the minmax problem

$$g \;=\; \arg\min_{g\in\mathcal{G}} \Big(\max_{d\in\mathcal{D}} V(d,g)\Big). \quad (2)$$

The following theorem, which generalizes Proposition 1 and Theorem 1 in Goodfellow et al. (2014a), shows that the max and (arg)min in (2) are well-defined. Note that our formulation neither requires g to be differentiable, nor $\mathcal{X}$ to be continuous.

Theorem 2.2. *Suppose* (A1)-(A2) *hold. Then*

$$V(d,g) \;=\; \int_{\mathcal{X}} \Big[\log(d(x))p(x) + \log(1-d(x))p^{(g)}(x)\Big]\mathrm{d}\mu(x) \quad (3)$$

for all $d \in \mathcal{D}$ and $g \in \mathcal{G}$. Hence, for any fixed $g \in \mathcal{G}$, any $d \in \mathcal{D}$ which maximizes $V(g,d)$ has the form

$$d(x) \;=\; \frac{p(x)}{p(x)+p^{(g)}(x)} \quad (4)$$

for μ-almost every $x \in \mathcal{X}$, implying that

$$\max_{d\in\mathcal{D}} V(g,d) \;=\; \int_{\mathcal{X}} \Big[p(x)\log\frac{p(x)}{p(x)+p^{(g)}(x)} + p_g(x)\log\frac{p^{(g)}(x)}{p(x)+p^{(g)}(x)}\Big]\mathrm{d}\mu(x). \quad (5)$$

Assuming that (A3)-(A4) *also hold, any generator $g \in \mathcal{G}$ that minimizes* (5) *is such that $p^{(g)} = p$ μ-almost everywhere, and* $\min_{g\in\mathcal{G}}\max_{d\in\mathcal{D}} V(g,d) = -\log(4)$.

The next theorem shows that the optimal discriminator d in (4) can be "perfect" if the overlap of the two distributions p and $p^{(g)}$ has measure zero. It generalizes Theorem 2.1 and 2.2 in Arjovsky and Bottou (2017), which are stated for the special case of $\mathcal{P}$ and $\mathcal{P}^{(g)}$ being not-perfectly-aligned submanifolds of $\mathbb{R}^k$.

Theorem 2.3. *Suppose* (A1)-(A3) *hold. For fixed $g \in \mathcal{G}$, let $\mathcal{P}, \mathcal{P}^{(g)} \subset \mathcal{X}$ be such that $\{x \in \mathcal{X} \,|\, p(x) > 0\} \subset \mathcal{P}$ and $\{x \in \mathcal{X} \,|\, p^{(g)}(x) > 0\} \subset \mathcal{P}^{(g)}$. Suppose $\mu(\mathcal{P} \cap \mathcal{P}^{(g)}) = 0$, $\mu(\partial(\mathcal{P} \setminus \mathcal{P}^{(g)})) = 0$ (where $\partial(\cdot)$ denotes the topological boundary) and $\mu(\partial(\mathcal{P}^{(g)} \setminus \mathcal{P})) = 0$. Then the optimal d in* (4) *satisfies $\mathbb{P}(d(X) = 1) = 1$ and $\mathbb{P}(d(g(Z)) = 0) = 1$. Moreover, without loss of generality, d is continuous μ-almost everywhere and, in the special case $\mathcal{X} = \mathbb{R}^k$, the gradient $\nabla d(x)$ exists and $\nabla d(x) = 0$ for μ-almost every $x \in \mathcal{X}$.*

In practice, the discriminator being constant on $\mathcal{P}$ and $\mathcal{P}^{(g)}$ poses problems. In particular, when g is fixed and d is trained till optimality, the gradients $\nabla d(x)$ may vanish and further updates of g may become impossible. In their Lemma 1 and 2, Arjovsky and Bottou (2017) establish that this is almost surely going to occur when the dimension of $\mathcal{Z}$ is smaller than the dimension of $\mathcal{X}$, and g is parameterized by a standard neural network. As we show next, it is more directly an inevitable consequence of using an empirical version of the objective (1) in practical GAN training.

2.3 GAN Training – Empirical Case

Let X_n^* be a random variable following the empirical distribution of $X_1, \ldots, X_n$. By $\mathbb{I}(\cdot)$ we denote the indicator function which evaluates to 1 if the statement in brackets is true, and to 0 otherwise. Note that, conditionally on $X_1, \ldots, X_n$, the distribution of X_n^* is

$$\mathbb{P}(X_n^* \in A \,|\, X_1, \ldots, X_n) \;=\; \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i \in A)$$

for $A \in \mathcal{A}$, and an analogous statement holds for the distribution of $g(Z_n^*)$ conditional on $Z_1, \ldots, Z_n$. It is important to note that practical GAN training (such as in Algorithm 1 in Goodfellow et al. (2014a)) is *not* with respect to the theoretical objective (1), but with respect to its empirical counterpart

$$V_n(d,g) := \mathbb{E}\big[\log(d(X_n^*)) \,|\, X_1, \ldots, X_n\big] + \mathbb{E}\big[\log(1-d(g(Z_n^*))) \,|\, Z_1, \ldots, Z_n\big]. \quad (6)$$

It appears there has been a wideheld belief among GAN practitioners that optimizing $V_n(d,g)$ leads to discriminators and generators with the same properties as stated in Theorem 2.3. As the following theorem shows, this isn't true in general. We add subscripts d_n and g_n to emphasize the dependency of discriminator and generator on the sample size n.

Theorem 2.4. *Suppose* (A1)-(A4) *hold. For fixed* $g \in \mathcal{G}$*, any* $d_n \in \mathcal{D}$ *maximizing* $V_n(d, g)$ *in* (6) *has the form*

$$d_n(x) = \frac{\sum_{i=1}^n \mathbb{I}(X_i = x)}{\sum_{i=1}^n \mathbb{I}(X_i = x) + \sum_{i=1}^n \mathbb{I}(g(Z_i) = x)} \quad (7)$$

for $x \in \{X_1, \ldots, X_n\} \cup \{g(Z_1), \ldots, g(Z_n)\}$ *(for all other* $x \in \mathcal{X}$*, the value* $d_n(x)$ *is arbitrary). If the cardinality of* $\{Z_1, \ldots, Z_n\}$ *is greater than or equal to the cardinality of* $\{X_1, \ldots, X_n\}$*, then any* $g_n \in \mathcal{G}$ *minimizing* (6) *for* $d = d_n$ *is such that* $\{g_n(Z_1), \ldots, g_n(Z_n)\} = \{X_1, \ldots, X_n\}$*.*

The first insight from Theorem 2.4 is a remarkably simple explanation for vanishing gradients in GAN training: Theorem 2.3 states general conditions under which the gradient of the generator may vanish when the discriminator is optimal. Theorem 2.4 shows that, for gradients with respect to the empirical training objective (6), these conditions are satisfied unless any of the training samples X_i and generated samples $g(Z_i)$ coincide. A sufficient condition for avoiding such coincidence is that both X and Z are continuous, and g is not constant on any subset A of Z for which $g(A)$ has strictly positive measure μ.

The second insight is that, when training with respect to (6), there is no theoretical guarantee that $p^{(g_n)} = p$ μ-almost everywhere for the optimal generator g_n – which contradicts Proposition 2 in Goodfellow et al. (2014a). The only guarantee is that, when applied to $Z_1, \ldots, Z_n$, g_n should reproduce the training samples $X_1, \ldots, X_n$. Note: this does not imply that g_n will *solely* reproduce training samples; in theory, the samples generated on $\mathcal{Z} \setminus \{Z_1, \ldots, Z_n\}$ are arbitrary. Hence, in contrary to the reasoning in Metz et al. (2016) and Arjovsky et al. (2017), the optimal g_n is *not* necessarily a Dirac function at the $x \in \mathcal{X}$ to which d_n assigns the highest values.

In practice, these undesirable properties could be mitigated for the following reasons: 1) the discriminator and generator function spaces $\mathcal{D}$ and $\mathcal{G}$ have limited capacity, hence the properties of d_n and g_n may only hold approximately; 2) similarly, alternate training of the generator and discriminator, or not training till optimality could alter the form of d_n and g_n, thereby circumventing pathologies.

2.4 Smoothing the Training Objective

A natural approach to avoid the issues pointed out in Theorem 2.3 and Theorem 2.4 is to smooth the Jensen-Shannon Divergence (JSD) in the training objective by adding noise to the input distributions of the optimal discriminator. We use the following assumption:

(A5) In addition to (A3), $(\mathcal{X}, +)$ is a topological group.

This allows us to consider the convolutions $p * p^{(\epsilon)}$ and $p^{(g)} * p^{(\epsilon)}$, which are the μ-densities of $X + \epsilon$ and $g(Z) + \epsilon$, respectively. The idea is to use, instead of the discriminator in (4), a modified version

$$d^*(x) = \frac{p * p^{(\epsilon)}(x)}{p * p^{(\epsilon)}(x) + p^{(g)} * p^{(\epsilon)}(x)}. \quad (8)$$

If the support of $p^{(\epsilon)}$ is sufficiently large, then the supports of $p * p^{(\epsilon)}$ and $p^{(g)} * p^{(\epsilon)}$ will overlap. Hence, it is not possible to construct an optimal d^* with the properties in Theorem 2.3. On the other hand, by the same arguments as in Theorem 2.2, the generator g minimizing (8) is such that $p^{(g)} * p^{(\epsilon)} = p * p^{(\epsilon)}$ μ-almost everywhere, which implies $p^{(g)} = p$ μ-almost everywhere, i.e. the optimal generator $g(Z)$ with respect to the theoretical objective still recovers the distribution of X. Next, we derive the form of the optimal discriminator for the modified empirical objective.

Theorem 2.5. *Suppose* (A1)-(A5) *hold and let* $g \in \mathcal{G}$ *be fixed. If we replace* X_n^* *and* $g(Z_n^*)$ *in* (6) *by* $X_n^* + \epsilon$ *and* $g(Z_n^*) + \epsilon$*, respectively, then the discriminator minimizing the objective has the form*

$$d_n^*(x) = \frac{\sum_{i=1}^n p^{(\epsilon)}(x - X_i)}{\sum_{i=1}^n p^{(\epsilon)}(x - X_i) + \sum_{i=1}^n p^{(\epsilon)}(x - g(Z_i))} \quad (9)$$

for $x \in \mathcal{X}$*. Same as in Theorem 2.4, if the cardinality of* $\{Z_1, \ldots, Z_n\}$ *is greater than or equal to the cardinality of* $\{X_1, \ldots, X_n\}$*, then any* $g_n^* \in \mathcal{G}$ *minimizing the objective* (6) *for* $d = d_n^*$ *is such that* $\{g_n^*(Z_1), \ldots, g_n^*(Z_n)\} = \{X_1, \ldots, X_n\}$*.*

Note that the smoothing of distributions outlined here is *not* equivalent to adding noise to the samples X_1, $\ldots$, X_n or $g(Z_1)$, $\ldots$, $g(Z_n)$ before optimizing the empirical objective, which would lead to the same result as in (7). In fact, ϵ is never sampled explicitly; it is only involved here through the computation of the convolved densities.

As Theorem 2.5 shows, smoothing the empirical distributions X_n^* and $g(Z_n^*)$ results in an optimal discriminator d_n^* which, if the support of $p^{(\epsilon)}$ is sufficiently large, won't cause vanishing gradients. However, there is still no guarantee that the optimal generator $g_n^*(Z)$ recovers the distribution of X apart from reproducing training samples. In the following section we discuss a new training objective which addresses this issue.

2.5 Kernel GANs

For all of further analyses we assume $\mathcal{X} = \mathbb{R}^k$, μ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^k$, $\mathcal{Z} \subset \mathbb{R}^l$, and $\Theta \subset \mathbb{R}^m$ for some $k, l, m \in \mathbb{N}$. Moreover, we assume that $\mathcal{G}$ is parameterized by $\theta \in \Theta$. We write g_θ for the generator parameterized by θ, and $p^{(\theta)}$ for the density of $g_\theta(Z)$.

It is instructive to note the resemblance of the optimal discriminator in (9) to a ratio of kernel densities: Let $K : \mathcal{X} \to \mathbb{R}$ be a measurable, bounded and square-integrable function (**kernel**), and $\sigma > 0$ (**bandwidth**). For $x \in \mathcal{X}$ consider the kernel density estimates

$$\hat{p}_{n,\sigma}(x) \quad := \quad \frac{1}{\sigma^k n} \sum_{i=1}^n K\left(\frac{x - X_i}{\sigma}\right), \tag{10}$$

$$\hat{p}^{(\theta)}_{n,\sigma}(x) \quad := \quad \frac{1}{\sigma^k n} \sum_{i=1}^n K\left(\frac{x - g_\theta(Z_i)}{\sigma}\right) \tag{11}$$

of $p(x)$ and $p^{(\theta)}(x)$. Choosing $K = p^{(\epsilon)}$, we can regard (9) as a kernel estimate of the density ratio $p(x)/(p(x)+p^{(\theta)}(x))$. Our key idea is to plug the optimal discriminator (9) back into the empirical training objective (6), i.e., consider $V_n(d, g)$ with $d = d_n^*$. This results in the **Kernel GAN** training objective:

$$\begin{aligned} K_n(\theta, \sigma, \varphi) &:= \\ &\frac{1}{n}\sum_{i=1}^n \log \frac{\hat{p}_{n,\sigma}(X_i) + \varphi}{\hat{p}_{n,\sigma}(X_i) + \hat{p}^{(\theta)}_{n,\sigma}(X_i) + 2\varphi} \\ &+ \frac{1}{n}\sum_{i=1}^n \log \frac{\hat{p}^{(\theta)}_{n,\sigma}(g_\theta(Z_i)) + \varphi}{\hat{p}_{n,\sigma}(g_\theta(Z_i)) + \hat{p}^{(\theta)}_{n,\sigma}(g_\theta(Z_i)) + 2\varphi} \end{aligned} \tag{12}$$

where $\varphi \geq 0$ is a regularizer to avoid underflow issues. In contrast to conventional GAN training, only the generator is explicitly updated when optimizing (12). The discriminator d_n^* is updated implicitly through changes in the density estimates (11). Note that plugging the optimal discriminator in (9) into the training objective (6) can be regarded as unrolling the discriminator same as in Metz et al. (2016), where in our case the discriminator is unrolled to closed-form optimality.

The following theorem establishes convergence of the objective (12).

Theorem 2.6. *Suppose* (A1)-(A5) *hold. Moreover, suppose p and $p^{(\theta)}$ are bounded and uniformly continuous for all $\theta \in \Theta$, and K has compact support and is of the form $K(x) = \phi(q(x))$, where q is a polynomial and ϕ a bounded non-negative function with bounded variation. Let $\sigma_n > 0$ be a sequence asymptotically equivalent to $Cn^{\frac{\delta-1}{k}}$ for some finite constant C and $\delta \in (0,1)$. Then*

$$\begin{aligned} \lim_{n\to\infty} K_n(\theta, \sigma_n, \varphi) &= \\ \int_{\mathcal{X}} \Big[&p(x) \log \frac{p(x) + \varphi}{p(x) + p^{(\theta)}(x) + 2\varphi} \\ &+ p^{(\theta)}(x) \log \frac{p^{(\theta)}(x) + \varphi}{p(x) + p^{(\theta)}(x) + 2\varphi}\Big] \mathrm{d}\mu(x) \end{aligned} \tag{13}$$

$\mathbb{P}$-almost surely for all θ and $\varphi > 0$.

The regularizer $\varphi > 0$ is required for establishing the convergence in (13). It results in estimates of the theoretical JSD that are asymptotically biased. In particular, while $K_n(\theta, \sigma_n, \varphi)$ converges to $-\log(4)$ (which is the minimum value of JSD) if θ is such that $p^{(\theta)} = p$, it may converge to smaller values for other values of θ. Hence, minimizing $K_n(\theta, \sigma_n, \varphi)$ would not result in a generator $g^{(\theta)}(Z)$ recovering X (although φ can be chosen arbitrarily small, hence the practical difference might be negligible). However, as we show in Appendix A.2, if $\mu(\mathcal{X}) < \infty$, then $K_n(\theta, \sigma_n, \varphi)$ can be modified such that its limit is minimized by a θ recovering the distribution of X.

3 EXPERIMENTS

In this section, we first demonstrate the practical learning of Kernel GANs on small and mid-sized datasets – a Mixture-of-Gaussian (MOG) toy dataset (Metz et al. (2016)) and MNIST (LeCun et al. (1998)). Further, we study the effect of kernel bandwidth along with practical approaches such as generating in a lower-dimensional feature space that is independently learned using an autoencoder.

Second, we establish practical usefulness of Kernel GANs by scaling them to two high-dimensional datasets: CIFAR-10 (Krizhevsky and Hinton (2009)) and CelebA (Liu et al. (2015)). We enable this with a modified training setup that involves kernel learning, similar to Li et al. (2017).

Finally, we perform various evalutions of the trained generators. In a quantitative evaluation, we compare Kernel GANs with MMD-based models (Li et al. (2015), Dziugaite et al. (2015), Li et al. (2017)), which also use kernel-based statistics, but in a non-adversarial fashion. Implementation details for all the experiments are stated in Appendix A.3 and A.4.

3.1 Learning Kernel GANs

Algorithm 1 outlines our general training protocol for learning the generator parameters θ that minimizes the training objective (12).

Algorithm 1 Training Protocol

```
Input: Training samples X, distribution of latent variable Z, initial kernel parameters σ, initial generator parameters θ, regularizer φ.
while stopping criterion not met do
    Sample a minibatch X_1, ..., X_n from X.
    Generate iid samples Z_1, ..., Z_n from Z.
    Update generator parameters θ according to gradients ∇_θ K_n(θ, σ, φ).
    Update kernel parameters σ.
end while
Output: Trained generator parameters θ.
```

Hyperparameters. Previously, Li et al. (2015) and Dziugaite et al. (2015) used RBF kernels in their training objectives for generative models. While Li et al. (2015) deploys a mixture of RBF kernels, Dziugaite et al. (2015) uses Bayesian Optimization to determine a suitable bandwidth. Moreover, both works suggest to use the median-trick (Gretton et al. (2012)) as a method to choose kernel bandwidths for computing MMD statistics. Intuitively, small bandwidths push the generator towards producing samples that are similar to the training set. However, initial bandwidths that are too small will not give gradients in areas that are far from the modes of the training set. We therefore explored gradual reductions of the bandwidth during training, similar to cooling schedules in simulated annealing (e.g., Hajek (1988), Nourani and Andresen (1998)). While the regularizer φ is required for deriving Theorem 2.6, we didn't find it to be critical in the experiments and therefore set it to zero. A further investigation of the practical effect of φ will be part of future work.

MOG Toy Dataset. For the MOG dataset, Z was a 100-dimensional standard normal distribution, and the generator was a three-layer fully connected network (128-relu-128-relu-128-tanh). Figure 1 shows the evolution of the generator during training, as the bandwidth σ is gradually decreased. Initially the generated samples $g(Z)$ are dispersed randomly. As the σ is decreased, they begin to concentrate around the modes of the MOG distribution.

MNIST. We succesfully trained three different generative models for MNIST. Two of these sampled directly in the space of (28×28) greyscale images. The third model used an autoencoder to map the images onto a lower-dimensional feature space, in which the generator was trained.

The three models used following architectures: a fully connected network (**FC**); a deconvolutional network with batch normalisation (**DC**); a fully connected network for the feature space (**FC-FS**). We adopted the architectures proposed in Li et al. (2015) for FC and FC-FS, and the one proposed in Radford et al. (2015a) for DC. As latent variable Z, all models used samples from a 10-dimensional uniform distribution. We use a mixture of RBF kernels for training these models (see the appendix for details). Generated samples are shown in Figure 2 (a)-(c). We find that samples from FC-FS have a very smooth appearance. DC generates sharper samples than FC, but observes some artifacts. A study of sharpness of generated samples for different bandwidths is discussed in the appendix.

We observe that training randomly initialized networks can be numerically unstable for very small bandwidths, leading to artifacts in the produced images. For very large bandwidths, we find that the generator often collapses and produces undesired samples like mean images. We noticed, however, that the generator model recovered when we increased or decreased the bandwidth appropriately in subsequent training iterations. An analysis is provided in the appendix. This suggests that kernel bandwidths can be used as "knobs" for correcting over- or underfitting of generative models during the training process.

3.2 Scaling Kernel GANs

Although sufficient in theory, we found it difficult to train Kernel GANs with plain RBF kernels for colored images. To impose more structure in kernel-based training, Li et al. (2015) suggest to use convolutional autoencoders that learn a lower dimensional feature space for images. Li et al. (2017) use this approach for training GMMN models on colored images but note quality issues in the generated samples. Instead they propose to learn a network which transforms the original space into a lower dimensional space over which the kernel is computed. We adopt their approach, leading to a modified Kernel GAN training objective, $K_n(\psi, \theta, \sigma, \varphi)$, which is the same as (12), except that the kernels operate on the space $f_\psi(\mathcal{X})$ instead of $\mathcal{X}$:

$$\hat{p}^{(\psi)}_{n,\sigma}(x) := \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{f_\psi(x) - f_\psi(X_i)}{\sigma}\right), \tag{14}$$

$$\hat{p}^{(\theta,\psi)}_{n,\sigma}(x) := \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{f_\psi(x) - f_\psi(g_\theta(Z_i))}{\sigma}\right). \tag{15}$$

The parameters $\{\theta, \psi\}$ are learned in a min-max fashion: $\min_\theta \max_\psi K_n(\psi, \theta, \sigma, \varphi)$. Similar to conventional GAN training, ψ and θ are optimized alternatingly. In practice additional regularization is required for stable learning. We use the experimental setup of Li et al. (2017), which models the function f_ψ as the encoder of

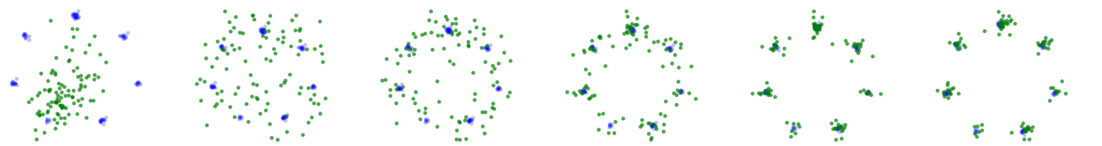

Figure 1: MOG toy dataset. Blue: training points. Green: samples produced by the generator. The *leftmost* figure is for the initial generator. Training phases for bandwidths (*left* to *right*): $0.8, 0.4, 0.2, 0.1, 0.05, 0.025$ (10,000 iterations were performed for each σ).

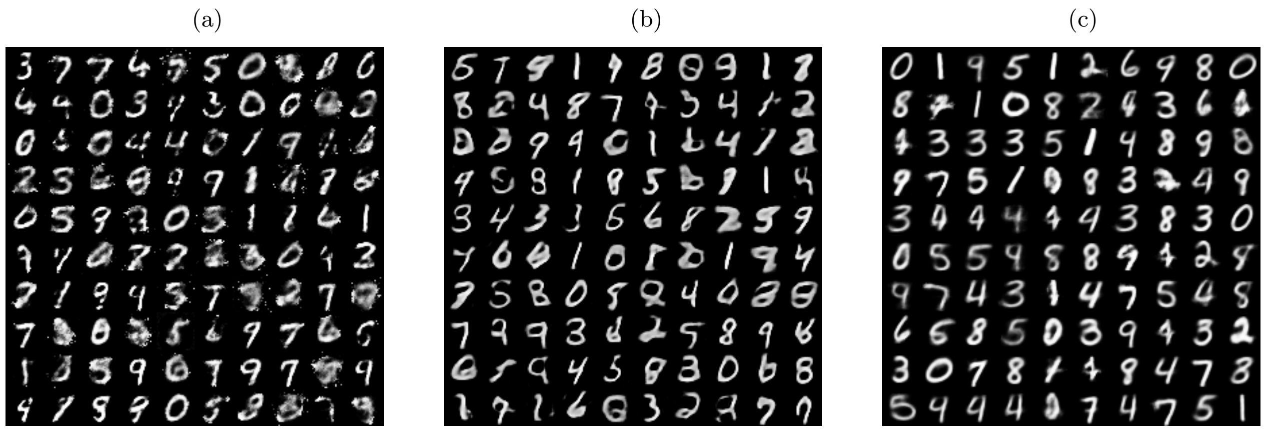

Figure 2: Training of generators for MNIST. (a): Fully connected network (FC). (b): Deep convolutional architecture (DC). (c): Fully connected network in feature space (FC-FS).

an autoencoder and regularizes the objective function with the autoencoder reconstruction loss.

CIFAR10 and CelebA. We train Kernel GANs for the CIFAR10 and CelebA datasets. We adopt and appropriately rescale hyperparameters and regularization weights of Li et al. (2017). We train a Deep Convolutional architecture for both datasets. The dimension of the encoded space $f\psi(\mathcal{X})$ is fixed to 100. CIFAR10 is trained with a 128-dimensional standard normal distribution for Z, and CelebA with a 64-dimensional Z. Samples obtained from the generators are shown in Figure 3. We find that they were qualitatively comparable to the results in Li et al. (2017).

3.3 Quantitative Evaluation

MNIST. Quantifying the performance of generative networks – particularly their ability to generalize and produce diverse samples – remains a challenging task (Theis et al. (2015); Wu et al. (2017)). In this paper, we report the following metrics:

Expected entropy (**EE**): Similar to Salimans et al. (2016), we train a probabilistic classifier (LeNet (LeCun et al.)) and compute the expected entropy of the predicted probabilities for samples from $g(Z)$. For all metrics, we use Monte-Carlo estimates of expected values, based on 10,000 samples. Expected nearest-neighbour distance (**ENN**): To assess the similarity of generated samples with the training set, we determine the expected value of the Euclidean distance between samples from $g(Z)$ and their nearest neighbor in the train set. LeNet score (**LS**): Motivated by the Inception score (Salimans et al., 2016), we compute the exponential of the expected Kullback-Leibler divergence between the predicted class probabilities for samples $g(Z)$, and the frequency of classes (=digits) in the MNIST train set. Jensen-Shannon divergence (**JSD**): We estimate the JSD between the data distribution and $g(Z)$ by computing (12) over the MNIST test set and samples from $g(Z)$. We report the corresponding values **JSD-F** and **JSD-S** of the first and second term in (12). Maximum Mean Discrepancy (**MMD**): Finally, we also report the MMD statistic (Gretton et al. (2012)).

Table 1 shows a comparison of different generators. **GMMN** and **GMMN-AE** are the data- and code-space Generative Moment Matching Networks proposed in Li et al. (2015). The numbers in the *MNIST Test* column are obtained by using the MNIST test set

Figure 3: Kernel GAN generated samples. (a): CIFAR10. (b): CelebA.

Table 1: Quantitative evaluation of different generators on MNIST

	FC	DC	FC-FS	GMMN-AE	GMMN	MNIST Test
EE	0.408	0.289	0.365	0.361	0.293	0.023
ENN	23.005	20.713	14.674	14.730	27.852	19.299
LS	6.601	7.464	6.916	6.948	7.408	9.752
JSD	-1.384	-1.385	-1.371	-1.372	-1.383	-1.386
JSD-F	-0.740	-0.740	-0.769	-0.776	-0.689	-0.693
JSD-S	-0.644	-0.645	-0.603	-0.596	-0.694	-0.693
MMD	0.019	0.017	0.084	0.091	0.005	0.000

instead of generated samples; they can be regarded as the performance of an ideal generator with optimal trade-off between fidelity (EE), diversity (ENN, LS), and overall consistency (JSD). In this regard, we find that DC performs best among all trained generators: it achieves the lowest EE, comparable ENN, and closest LS in comparison with *MNIST Test*. FC-FS and GMMN-AE achieve high fidelity, but exhibit less diversity. Interestingly, the two term of the JSD are imbalanced for these models. We hypothesize that keeping JSD-F and JSD-S balanced during training is key to obtaining generators with good generalization capacity.

CIFAR10. We compute the Inception score (Salimans et al. (2016)) mean and standard deviation for $5\times$ 10k samples from a Kernel GAN which is trained for 5,000 iterations. The score for held-out CIFAR10 images (which can be regarded as gold standard) is 11.95 ($\pm$ .20). Kernel GAN yields a score of 4.22 ($\pm$ .02), which is significantly higher than the scores of GMMN-AE and GMMN (3.94 $\pm$ .04 and 3.47 $\pm$ .03, respectively), but lower than the ones for MMD-GAN and WGAN (6.17 $\pm$ .07 and 5.88 $\pm$ .07 respectively, (Li et al., 2017)). We expect that optimizing hyperparameters and regularization weights for Kernel GANs can yield improved scores.

4 CONCLUSIONS

We established a rigorous framework for analyzing statistical properties of Generative Adversarial Network training. To overcome potential pathologies (in particular, vanishing gradients), we introduced a novel training objective, which can be regarded as minimizing a non-parametric estimate of the Jensen-Shannon Divergence. We analyzed its asymptotic properties and showed its practical applicability.

We see several directions for future work: 1) Advance the design of optimal kernels and strategies for annealing the bandwidths. 2) Further analyze statistical properties of the proposed training objective, in particular, the effect of the regularizer. 3) Investigate the effect of imbalances between the first and second term in the training objective; we believe this could lead to the design of adaptive training protocols which ensure both fidelity and diversity of generator samples. 4) Investigate how the learning of the feature mapping f_ψ proposed in Section 3.2 affects the theoretical guarantees.

References

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573v2*, 2017.

Vladimir I. Bogachev. *Measure Theory*. Springer Verlag, Berlin, 2007.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengion. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, pages 258–267, Arlington, Virginia, United States, 2015. AUAI Press. ISBN 978-0-9966431-0-8. URL `http://dl.acm.org/citation.cfm?id=3020847.3020875`.

Evarist Giné and Armelle Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. *Annales de l'Insitut Henri Poincaré*, 38 (6):907–921, 2002.

Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Wade-Farley, Sherjil Ozair, and Aaron Courville. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27:2672–2680, 2014a.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

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

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028v1*, 2017.

Bruce Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13(2):311–329, 1988.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998.

Yann LeCun et al. Lenet-5, convolutional neural networks.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *arXiv preprint arXiv:1705.08584*, 2017.

Yujia Li, Kevin Swersky, and Richard S Zemel. Generative moment matching networks. In *ICML*, pages 1718–1727, 2015.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *CVPR*, pages 3730–3738, 2015.

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483v4*, 2016.

Yaghout Nourani and Bjarne Andresen. A comparison of simulated annealing cooling strategies. *J. Phys. A: Math. Gen.*, 31:8373–8385, 1998.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015a.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015b.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 29:2226–2234, 2016.

Alan Henry Schoenfeld. Continuous measure-preserving maps onto peano spaces. *Pacific Journal of Mathematics*, 58(2):627–642, 1975.

Lucas Theis, Aäron Van Den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.

Ilya O. Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. Adagan: Boosting generative models. *arXiv preprint arXiv:1701.02386*, 2017.

Stephen Willard. *General Topology*. Dover Publications, Mineola, New York, 1970.

Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. *5th International Conference on Learning Representations*, 2017.