Ratio Matching MMD Nets: Low dimensional projections for effective deep generative models

Akash Srivastava
School of Informatics
University of Edinburgh
akash.srivastava@ed.ac.uk

Kai Xu
School of Informatics
University of Edinburgh
kai.xu@ed.ac.uk

Michael U. Gutmann
School of Informatics
University of Edinburgh
Michael.Gutmann@ed.ac.uk

Charles Sutton
School of Informatics, Google Brain & The Alan Turing Institute
University of Edinburgh
csutton@inf.ed.ac.uk

Abstract

Deep generative models can learn to generate realistic-looking images on several natural image datasets, but many of the most effective methods are adversarial methods, which require careful balancing of training between a generator network and a discriminator network. Maximum mean discrepancy networks (MMD-nets) avoid this issue using the kernel trick, but unfortunately they have not on their own been able to match the performance of adversarial training. We present a new method of training MMD-nets, based on learning a mapping of samples from the data and from the model into a lower dimensional space, in which MMD training can be more effective. We call these networks ratio matching MMD networks (RM-MMDnets). We train the mapping to preserve density ratios between the densities over the low-dimensional space and the original space. This ensures that matching the model distribution to the data in the low-dimensional space will also match the original distributions. We show that RM-MMDnets have better performance and better stability than recent adversarial methods for training MMD-nets.

1 Introduction

Deep generative models (Goodfellow et al., 2014; Arjovsky et al., 2017) have been shown to learn to generate realistic-looking images on several natural image datasets. These methods train a deep neural network, called a generator, to transform samples from a noise distribution to samples from the data distribution. Most methods use adversarial learning (Goodfellow et al., 2014), in which the generator is pitted against a critic function, also called a discriminator, which is trained to distinguish between the samples from the data distribution and from the generator. Upon successful training the two sets of samples become indistinguishable with respect to the critic.

Maximum mean discrepancy (MMD) networks (Li et al., 2015; Dziugaite et al., 2015) are a class of generative models that are trained to minimize the MMD between the true data distribution and the model distribution. MMD networks are similar in spirit to generative adversarial networks (GANs) (Goodfellow et al., 2014), in the sense that the MMD is defined by maximizing over a class of critic functions. However, in contrast to GANs, where finding the right balance between generator and critic is difficult, training is simpler for MMD networks because using the kernel trick the MMD can be estimated in closed form without the need to numerically optimize over critic functions. This avoids the need in GANs to numerically solve a saddlepoint problem.

Preprint. Work in progress.

Unfortunately, although MMD networks work well on low dimensional data, these networks have not on their own matched the performance of adversarial methods on higher dimensional datasets, such as natural images (Dziugaite et al., 2015). Several authors (Li et al., 2017; Bińkowski et al., 2018) suggest that a reason is that MMD is sensitive to the choice of kernel. Li et al. (2017) propose a method called MMD-GAN, in which the critic maps the samples from the generator and the data into a lower-dimensional representation, and MMD is applied in this transformed space. This can be interpreted as a method for learning the kernel in MMD. The critic is learned adversarially by maximizing the MMD at the same time as it is minimized with respect to the generator. This is much more effective than MMD networks, but training MMD-GANs can be challenging, because the need to balance training of the learned kernel and the generator can create a sensitivity to hyperparameter settings. In practice, it is necessary to introduce several additional penalties to the loss function in order for training to be effective.

In this work, we present a novel training method for MMD networks based on a new principle for optimizing the critic. Like previous work, our goal is for the critic to map the samples into a lower-dimensional space in which the MMD network estimator will be more effective. Our proposal is that the critic should preserve density ratios, namely, the ratio of the true density to model density should be preserved under the mapping defined by the critic. If the critic is successful in this, then matching the generator to the true data in the lower dimensional space will also match the distributions in the original space. We call networks that have been trained using this criterion *ratio matching MMD networks (RM-MMDnets).* This proposal builds on previous work by Sugiyama et al. (2011) that considered *linear* dimensionality reduction for density ratio estimation. We show empirically that our method is not only able to generate high quality images but by virtue of being non-adversarial it avoids saddlepoint optimization and hence is more stable to train and robust to the choice of hyperparameters.

2 Background and Related Work

Given data $x_i \in \mathbb{R}^D$ for $i \in \{1 \dots N\}$ from a distribution of interest with density p_x, the goal of deep generative modeling is to learn a parametrized function $G_\gamma : \mathbb{R}^h \mapsto \mathbb{R}^D$, called a generator network, that maps samples from a noise distribution p_z to samples from the model distribution. Since G_γ defines a new random variable, we denote its density function by q_x, and also write $x^q = G_\gamma(z)$, where we suppress the dependency on γ. The parameters γ of the generator are chosen to minimize a loss criterion which encourages q_x to match p_x.

2.1 Maximum Mean Discrepancy

Maximum mean discrepancy measures the discrepancy between two distributions as the maximum difference between the expectations of a class of functions $\mathcal{F}$, that is,

$$\mathrm{MMD}_{\mathcal{F}}(p,q) = \sup_{f \in \mathcal{F}} \mathbb{E}_p[f(x)] - \mathbb{E}_q[f(x)], \tag{1}$$

where $\mathbb{E}$ denotes expectation. If $\mathcal{F}$ is chosen to be a rich enough class, then $\mathrm{MMD}(p,q) = 0$ implies that $p = q$. Gretton et al. (2012) show that it is sufficient to choose $\mathcal{F}$ to be a unit ball within a reproducing kernel Hilbert space $\mathcal{R}$ with kernel k. Given samples $x_1 \dots x_N \sim p$ and $y_i \dots y_M \sim q$, we can estimate $\mathrm{MMD}_{\mathcal{R}}$ as

$$\mathrm{M\hat{M}D}_{\mathcal{R}}(p,q) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N} k(x_i, x_{i'}) - \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k(x_i, y_j) + \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j'=1}^{M} k(y_j, y_{j'}). \tag{2}$$

2.2 MMD networks and MMD-GANs

Li et al. (2015) and Dziugaite et al. (2015) independently proposed MMD networks, which use the MMD criterion to train a deep generative model. Unlike f-divergences, MMD is well defined even for distributions that do not have an overlapping support, which is an important consideration for training generative models (Arjovsky et al., 2017). Therefore, MMD networks use (2) in order to minimize the discrepancy between the distributions q_x and p_x with respect to G_γ. However, the sample quality of MMD networks generally degrades for higher dimensional or color image datasets (Li et al., 2015).

To address this problem, Li et al. (2017) introduce MMD-GANs, which use a critic $f_\theta : \mathbb{R}^D \mapsto \mathbb{R}^K$ to map the samples to a lower dimensional space $\mathbb{R}^K$, and train the generator to minimize MMD in this reduced space. Essentially this corresponds to learning the kernel function for MMD, because if f_θ is injective and k_0 is a kernel in $\mathbb{R}^K$, then $k(x, x') = k_0(f_\theta(x), f_\theta(x'))$ is a kernel in $\mathbb{R}^D$. This injectivity constraint on f_θ is imposed by introducing another deep neural network, f'_ϕ which is trained to approximately invert f_θ using an auto-encoding penalty. The critic f_θ is trained using an adversarial criterion, but this then requires numerical saddlepoint optimization, and avoiding this was one of the main attractions of MMD in the first place.

Successfully training f_θ in practice required a penalty term called feasible set reduction on the class of functions that f_θ can learn to represent. Defining $\bar{p}$ and $\bar{q}$ respectively as the distributions of the random variables obtained by applying f_θ to p_x and q_x, the training criteria for MMD-GANs are

$$\begin{aligned}\mathcal{L}(\theta, \phi) &= \mathrm{MMD}\Big[\bar{p}\big(f_\theta(x)\big), \bar{q}\big(f_\theta(G_\gamma(z))\big)\Big] - \lambda_1 d\Big[x, f'_\phi\big(f_\theta(G_\gamma(z))\big)\Big] \qquad (3)\\ &\quad + \lambda_2 \min\Big[\mathbb{E}[f_\theta(x)] - \mathbb{E}[f_\theta(G_\gamma(z))], 0\Big]\\ \mathcal{L}(\gamma) &= \mathrm{MMD}\Big[\bar{p}\big(f_\theta(x)\big), \bar{q}\big(f_\theta(G_\gamma(z))\big)\Big] + \lambda_3 \min\Big[\mathbb{E}[f_\theta(x)] - \mathbb{E}[f_\theta(G_\gamma(z))], 0\Big],\end{aligned}$$

where $x \sim p_x$ and $z \sim p_z$ are samples from their respective distributions. The function d denotes an expected auto-encoding penalty that ensures that f is approximately injective. Furthermore, f is restricted to be k-Lipschitz continuous by using a low learning rate and explicitly clipping the gradients during update steps of f akin to WGAN (Arjovsky et al., 2017).

Our work is similar in spirit to MMD-GANs, in that we will also learn a critic function to improve the performance of MMD networks. The main differences are that we will not use an adversarial criterion to learn f_θ, and that we do not require the function $k(f_\theta(\cdot), f_\theta(\cdot))$ to be a kernel function. These differences will greatly simplify our training algorithm, as we do not require an additional autoencoding penalty or feasible set reduction as in their method. We will also show (Section 4) that our method is more stable in training.

2.3 Dimensionality Reduction for Density Ratio Estimation

Sugiyama et al. (2011) suggest that density ratio estimation for distributions p and q over $\mathbb{R}^D$ can be more accurately done in lower dimensional subspaces $\mathbb{R}^K$. They propose to first learn a linear projection to a lower dimensional space by maximizing an f-divergence between the distributions $\bar{p}$ and $\bar{q}$ of the projected data and then estimate the ratio of $\bar{p}$ and $\bar{q}$. They showed that the projected distributions preserve the original density ratio. Our method builds on this insight, generalizing it to non-linear projections and incorporating it into a method for deep generative modeling.

3 Method

Our aim will be to enjoy the advantages of MMD networks, but to improve their performance by mapping the data into a lower-dimensional space, using a critic network f_θ, before computing the MMD criterion. Because MMD with a fixed kernel performs well for lower-dimensional data (Li et al., 2015; Dziugaite et al., 2015), we hope that by choosing $K < D$, we will improve the performance of the MMD network. Instead of training f_θ using an adversarial criterion like MMD-GAN, we aim at a more stable training method by introducing a different principle for training the critic.

The idea is to train f_θ so as to preserve density ratios between the original space and the lower-dimensional space. Let $\bar{q}$ be the density of the transformed simulated data, i.e., the density of the random variable $f_\theta(G_\gamma(z))$, where $z \sim p_z$. Similarly let $\bar{p}$ be the density of the transformed data, i.e., the density of the random variable $f_\theta(x)$. Then our goal is to choose θ so that p_x/q_x equals $\bar{p}/\bar{q}$. The motivation is that if density ratios are preserved by f_θ, then matching the generator to the data in the transformed space will also match it in the original space (Section 3.3). The capacity of f_θ should be chosen to strike a trade-off between dimensionality reduction and ability to approximate the ratio. If the data lie on a lower-dimensional manifold in $\mathbb{R}^D$, which is the case for e.g. natural images, then it is reasonable to suppose that we can find a critic that strikes a good trade-off.

Inspired by the work of Sugiyama et al. (2012), we train f_θ to minimize the average squared difference between these two density ratios (Section 3.1). To compute this criterion, we need to estimate density ratios $\bar{p}/\bar{q}$, which can be done in closed form using MMD (Section 3.2). Our method then alternates stochastic gradient descent (SGD) steps between training the critic and the generator. The generator is trained as an MMD network to match the transformed data $\{f_\theta(x_i)\}$ with transformed outputs from the generator $\{f(G_\gamma(z_i)\}$ in the lower dimensional space. These gradient steps are alternated with SGD steps on the the critic (Section 3.3).

3.1 Measuring Discrepancy

Our principle is to choose f_θ so that the resulting densities $\bar{p}$ and $\bar{q}$ preserve the density ratio between p_x and q_x. We will choose f_θ to minimize the distance between the two density ratio functions

$$r_x(x) = p_x(x)/q_x(x) \qquad r_\theta(x) = \bar{p}(f_\theta(x))/\bar{q}(f_\theta(x)).$$

One way to measure how well f preserves density ratios is to use the squared distance

$$D^*(\theta) = \int q_x(x)\left(\frac{p_x(x)}{q_x(x)} - \frac{\bar{p}(f_\theta(x))}{\bar{q}(f_\theta(x))}\right)^2 dx. \tag{4}$$

This objective is minimized only when the ratios match exactly, that is, when $r_x = r_\theta$ for all x in the support of q_x. Clearly a distance of zero can be trivially achieved if $K = D$ and if f_θ is the identity function. But nontrivial optima can exist as well. For example, suppose that p_x and q_x are "intrinsically low dimensional" in the following sense. Suppose $K < D$, and consider two distributions p_0 and q_0 over $\mathbb{R}^K$, and an injective map $T : \mathbb{R}^K \to \mathbb{R}^D$. Suppose that T maps samples from p_0 and q_0 to samples from p_x and q_x, by which we mean $p_x(x) = J(\mathbf{D}T)p_0(T^{-1}(x))$, and similarly for q_x. Here $J(\mathbf{D}T)$ denotes the Jacobian $J(\mathbf{D}T) = \sqrt{|\delta T \delta T^\top|}$ of T. Then we have that D^* is minimized to 0 when $f_\theta = T^{-1}$.

We proceed by expanding (4) as

$$D^*(\theta) = C + \int q_x(x)\left(\frac{\bar{p}(f_\theta(x))}{\bar{q}(f_\theta(x))}\right)^2 dx - 2\int q_x(x)\frac{p_x(x)}{q_x(x)}\frac{\bar{p}(f_\theta(x))}{\bar{q}(f_\theta(x))}dx, \tag{5}$$

where C does not depend on θ. Equation (5) can be minimized empirically using samples $x_1^q \dots x_N^q \sim q_x$, yielding the critic loss function

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}[r_\theta(x_i^q)]^2 - \frac{2}{N}\sum_{i=1}^{N}[r_x(x_i^q)r_\theta(x_i^q)]. \tag{6}$$

Optimizing this requires a way to estimate $r_x(x_i^q)$ and $r_\theta(x_i^q)$, which we present in the next section.

3.2 Density Ratio Estimation

In order to estimate the density ratios r_θ and r_x, we have several choices of estimators (Sugiyama et al., 2012). In our work, however, we employ the MMD criterion, because this allows a closed-form estimate. The MMD estimator of r_θ (Sugiyama et al., 2012) is given by optimizing

$$\min_{r\in\mathcal{R}}\left\|\int k(y;.)\bar{p}(y)dy - \int k(y;.)r(y)\bar{q}(y)dy\right\|_{\mathcal{R}}^2, \tag{7}$$

where k is a kernel function. It is easy to see that at the minimum, we have $r = \bar{p}/\bar{q}$. Notice that to compute (6), we need the values of r_θ and r_x only for the points $x_1^q \dots x_N^q$. In other words, we need to approximate the vectors $\mathbf{r}_{q,\theta} = [r_\theta(x_1^q) \dots r_\theta(x_N^q)]^T$ and $\mathbf{r}_{q,x} = [r_x(x_1^q) \dots r_x(x_N^q)]^T$. Following Sugiyama et al. (2012), we replace the integrals in (7) with averages over the points $x_1^q \dots x_N^q$ and over points $x_1^p \dots x_N^p \sim p_x$. The minimizing values of $\mathbf{r}_{q,\theta}$ can then be computed as

$$\hat{\mathbf{r}}_{q,\theta} = K_{q,q}^{-1}K_{q,p}\mathbf{1}. \tag{8}$$

Here $K_{q,q}$ and $K_{q,p}$ denote the Gram matrices defined by $[K_{q,q}]_{i,j} = k(f_\theta(x_i^q), f_\theta(x_j^q))$ and $[K_{q,p}]_{i,j} = k(f_\theta(x_i^q), f_\theta(x_j^p))$. A similar equation can be used to compute the estimate $\hat{\mathbf{r}}_{q,x}$. Substituting these estimates into (6), we get

$$\hat{\mathcal{L}}(\theta) = \frac{1}{N}\|\hat{\mathbf{r}}_{q,\theta}\|^2 - \frac{2}{N}(\hat{\mathbf{r}}_{q,x} \odot \hat{\mathbf{r}}_{q,\theta})^T\mathbf{1}, \tag{9}$$

where $\odot$ refers to elementwise multiplication. This objective is minimized to learn the critic f_θ.

Normalization. We add a correction to (9) to account for certain inaccuracies in the estimate $\hat{\mathbf{r}}_{q,\theta}$. For the true density ratio, we have $\int q_x(x) r_\theta(x) = 1$. However, we are not guaranteed that the vector $\hat{\mathbf{r}}_{q,\theta}$ is normalized, or even nonnegative. Indeed, we found it was common in practice for the estimated ratios to be negative. We can correct for this by optimizing $\hat{\mathcal{L}}$ under the constraint that $N^{-1}\mathbf{r}_{q,\theta}^T\mathbf{1} = 1$. In practice, we found it to be slightly more effective to estimate the inverse density ratio

$$r^{qp}(x) = \frac{\bar{q}(f_\theta(x))}{\bar{p}(f_\theta(x))}$$

at the points $x_1^p \dots x_L^p$, yielding an estimated vector of density ratios $\hat{\mathbf{r}}_\theta^{qp}$. Then if we take the Lagrangian with respect to the constraint $M^{-1}\hat{\mathbf{r}}_\theta^{qp\,T}\mathbf{1} = 1$ and remove constants, we obtain

$$\hat{\mathcal{L}}_0(\theta) = \frac{1}{N}\|\hat{\mathbf{r}}_{q,\theta}\|^2 - 2\frac{1}{N}(\hat{\mathbf{r}}_{q,x} \odot \hat{\mathbf{r}}_{q,\theta})^T\mathbf{1} - \frac{\lambda}{L}\hat{\mathbf{r}}_\theta^{qp\,T}\mathbf{1}, \tag{10}$$

where we take the Lagrange multiplier λ as a tuning parameter. Although λ does introduce a new regularization parameter, we found that the arbitrary choice of $\lambda = 2$ worked acceptably in practice, and we did not tune this parameter further. Essentially this normalization term discourages θ from choosing transformations under which the MMD ratio estimator would violate the normalization constraint.

Empirical Estimation: Estimation of r_x using MMD is difficult because the dimensionality of x is large, and we find empirically that a large number of samples are required for an accurate estimate. This also causes the estimate to be computationally demanding, because of the need to invert a large Gram matrix. But empirically we notice that the second term in (10) is almost always very small (on the order of $1e^{-6}$) and does not contribute towards the training of the critic (see appendix for empirical evidence). Therefore, in practice, dropping this term from objective, i.e. training f_θ according to

$$\hat{\mathcal{L}}_1(\theta) = \frac{1}{N}\|\hat{\mathbf{r}}_{q,\theta}\|^2 - \frac{\lambda}{L}\hat{\mathbf{r}}_\theta^{qpT}\mathbf{1}, \tag{11}$$

provides equivalent or better performance, so we use $\hat{\mathcal{L}}_1$ in our experiments. We were able to successfully train networks using $\hat{\mathcal{L}}_0$, but at much higher computational cost.

3.3 Generator Loss

To train the generator network G_γ, we minimize the MMD in the low-dimensional space, transforming both the generated data and the true data by f_θ. In other words, we minimize the MMD between $\bar{p}$ and $\bar{q}$. We sample points $z_1 \dots z_M \sim p_z$ from the input distribution of the generator. Then using the empirical estimate (2) of the MMD, we define the generator loss function as

$$\begin{aligned}\hat{\mathcal{L}}_2(\gamma) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N} k(f_\theta(x_i), f_\theta(x_{i'})) - \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k(f_\theta(x_i), f_\theta(G_\gamma(z_j))) \\ + \frac{1}{N^2}\sum_{j=1}^{M}\sum_{j'=1}^{M} k(f_\theta(G_\gamma(z_j)), f_\theta(G_\gamma(z_{j'}))),\end{aligned} \tag{12}$$

which we minimize with respect to γ for a fixed critic f_θ. Finally, the overall training proceeds by alternating SGD steps between $\hat{\mathcal{L}}_1$ and $\hat{\mathcal{L}}_2$. Unlike WGAN (Arjovsky et al., 2017) and MMD-GAN, we do not require the use of gradient clipping, feasible set reduction and autoencoding regularization terms from (3).

If we succeed in matching the generator to the true data in the low-dimensional space, then we have also matched the generator to the data in the original space, in the limit of infinite data. To see this, suppose that we have γ^* and θ^* such that $D^*(\theta^*) = 0$ and that $M_y = \text{MMD}(\bar{p}, \bar{q}) = 0$. Then for all x, we have $r_x(x) = r_{\theta^*}(x)$ because $D^*(\theta^*) = 0$, and that $r_{\theta^*}(x) = 1$, because $M_y = 0$. This means that $r_x(x) = 1$, so we have that $p_x = q_x$.

Table 1: Sample quality (measured by FID; lower is better) of RM-MMDnets compared to GANs.

Archtitecture	Dataset	MMD-GAN	GAN	RM-MMDnet
DCGAN	**Cifar10**	40 (0.56)	26.82 (0.49)	**24.85 (0.94)**
Small Critic	**Cifar10**	210.85 (8.92)	31.64 (2.10)	**24.82 (0.62)**
DCGAN	**CelebA**	41.105 (1.42)	30.97 (5.32)	**27.04 (4.24)**

Table 2: Performance of MMD-GAN (Inception scores; larger is better) for MMD-GAN with and without additional penalty terms: feasible set reduction (FSR) and the autoencoding loss (AE).

Batch Size	MMD+FSR+AE	MMD+FSR	MMD+AE	MMD
64	5.35 (0.05)	5.40 (0.04)	3.26 (0.03)	3.51 (0.03)
300	5.43 (0.03)	5.15 (0.06)	3.68 (0.22)	3.87 (0.03)

4 Experiments

In this section we describe the experiments that we conducted in order to establish the performance of our method. We compare RM-MMDnets against MMD-GANs and vanilla GANs, on the Cifar10 and CelebA image datasets. To evaluate the sample quality and resilience against mode dropping, we used Frechet Inception Distance (FID) (Heusel et al., 2017).[1] Like the Inception Score (IS), FID also leverages a pre-trained Inception Net to quantify the quality of the generated samples, but it is more robust to noise than IS and can also detect intra-class mode dropping (Lucic et al., 2017). FID first embeds both the real and the generated samples into the feature space of a specific layer of the pre-trained Inception Net. It further assumes this feature space to follow a multivariate Gaussian distribution and calculates the mean and covariance for both sets of embeddings. The sample quality is then defined as the Frechet distance between the two Gaussian distributions, which is

$$\text{FID}(x_p, x_q) = \|\mu_{x_p} - \mu_{x_q}\|_2^2 + \text{Tr}(\Sigma_{x_p} + \Sigma_{x_q} - 2(\Sigma_{x_p}\Sigma_{x_q})^{\frac{1}{2}}),$$

where $(\mu_{x_p}, \Sigma_{x_p})$, and $(\mu_{x_q}, \Sigma_{x_q})$ are the mean and covariance of the sample embeddings from the data distribution and model distribution. We report FID on a held-out set that was not used to train the models. We run all the models three times from random initializations and report the mean and standard deviation of FID over the initializations. To ensure that we are fairly comparing with Li et al. (2017), who report IS rather than FID, we computed IS values on the Cifar10 data set as well. See the appendix.

Architecture: We test all the methods on the same architectures for the generator and the critic, namely a four-layer DCGAN architecture (Radford et al., 2015), because this has been consistently shown to perform well for the datasets that we use. Additionally, to study the effect of changing architecture, we also evaluate a slightly weaker critic, with the same number of layers but half the number of hidden units. Details of the architectures are provided in the appendix.

Hyperparameters: To facilitate fair comparison with MMD-GAN we set all the hyperparameters shared across the three methods to the values used in Li et al. (2017). Therefore, we use a learning rate of $5e^{-5}$ and set the batch size to 64. For the MMD-GAN and RM-MMDnets, we used the same set of RBF kernels that were used in Li et al. (2017). We used the implementation of MMD-GANs from Li et al. (2017).[2] We leave all the hyper-parameters that are only used by MMD-GAN, namely the weights λ_1, λ_2, and λ_3 from the MMD-GAN objective (3), to the settings in the authors' code. For RM-MMDnets, we choose $K = h$, that is, the critic dimensionality equals the input dimensionality of the generator. We present an evaluation of hyperparameter sensitivity in Section 4.2.

4.1 Image Quality

We now look at how our method competes against GANs and MMD-GANs on sample quality and mode dropping on Cifar10 and CelebA datasets. Results are shown in Table 1. Clearly, RM-MMDnets

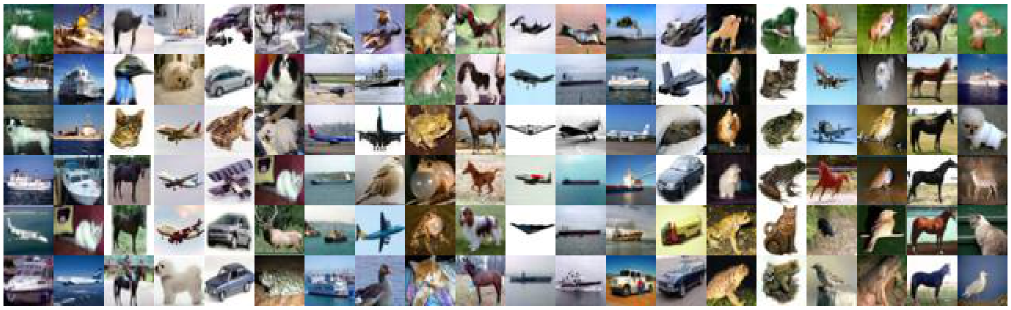
[1] We use a standard implementation available from `https://github.com/bioinf-jku/TTUR`

[2] Available at `https://github.com/OctoberChang/MMD-GAN`.

Table 3: Sample quality (FID) of fully convolutional architecture originally used for MMD-GAN by Li et al. (2017).

Architecture	**Dataset**	**MMD-GAN**
Fully Convolutional	**Cifar10**	38.39 (0.28)
Fully Convolutional	**CelebA**	40.27 (1.32)

Figure 1: Nearest training images to random samples from an RM-MMDnet trained on Cifar10. In each column, the top image is a sample from the generator, and the images below it are the nearest neighbors.



outperform both baselines. For CelebA, we do not run experiments using the weaker critic, because this is a much larger and higher-dimensional dataset, so a low-capacity critic is unlikely to work well.

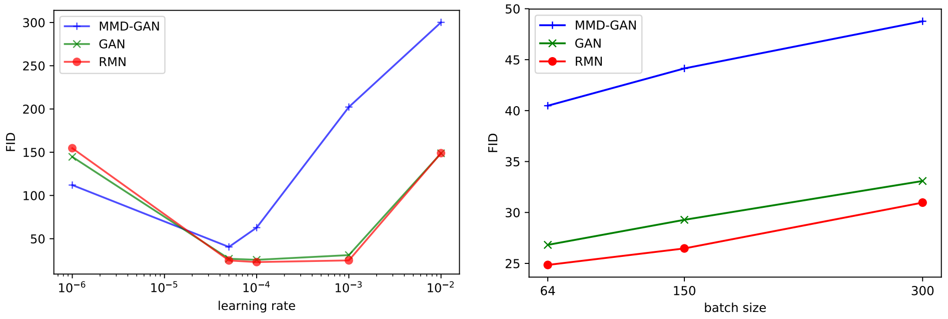
To provide evidence that RM-MMDnets are not simply memorizing the training set, we note that we measure FID on a held-out set, so a network that memorized the training set would be likely to have poor performance. We provide additional qualitative evidence that our method is not simply memorizing the dataset in Figure 1. This figure shows the five nearest neighbors from the training set for 20 randomly generated samples from the trained generator of our RM-MMDnet. None of the generated images have an exact copy in the training set, and qualitatively the 20 images appear to be fairly diverse.

Note that our architecture is different from that used in the results of Li et al. (2017). That work uses a fully convolutional architecture for both the critic and the generator, which results in an order of magnitude more weights. This makes a large comparison more expensive, and also risks overfitting on a small dataset like Cifar10. However, for completeness, and to verify the fairness of our comparison, we also report the FID that we were able to obtain with MMD-GAN on this fully-convolutional architecture in Table 3. Compared to our experiments using MMD-GAN to train the DCGAN architecture, the performance of MMD-GAN with the fully convolutional architecture remains unchanged for the larger CelebA dataset. On Cifar10, not surprisingly, the larger fully convolutional architecture performs slightly better than the DCGAN architecture trained using MMD-GAN. The difference in FID between the two different architectures is relatively small, justifying our decision to compare the generative training methods on the DCGAN architecture.

4.2 Sensitivity to Hyperparameters

GAN training can be sensitive to the learning rate (LR) and the batch size used for training (Lucic et al., 2017). We examine the effect of learning rates and batch sizes on the performance of all three methods. Figure 2a compares the performance as a function of the learning rates. We see that RM-MMDnets are much less sensitive to the learning rate than MMD-GAN, and are about as robust to changes in the learning rate as a vanilla GAN. MMD-GAN seems to be especially sensitive to this hyperparameter. We suggest that this might be the case since the critic in MMD-GAN is restricted to the set of k-Lipschitz continuous functions using gradient clipping, and hence needs lower learning rates. Similarly, Figure 2b shows the effect of the batch size on the three models. We notice that all models are slightly sensitive to the batch size, and lower batch size is in general better for all three methods.

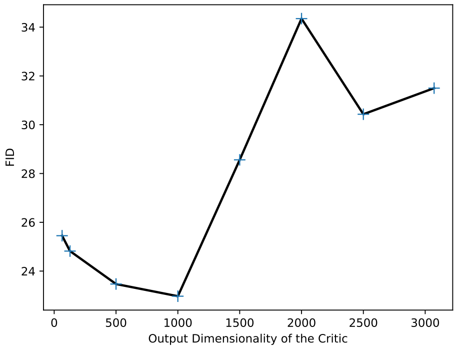
Figure 2: Hyper-parameter sensitivity of MMD-GAN, GAN and RMN on Cifar10 dataset.



(a) FID vs Learning Rate

(b) FID vs Batch Size

Figure 3: Sample quality (measured by FID) as a function of the output dimensionality of the critic.



4.3 Stability of MMD-GANs

For MMD-GANs, we evaluate the effect of the various stabilization techniques used for training, namely the autoencoder penalty (AE) and the feasible set reduction (FSR) techniques from (3) on the Cifar10 data over two settings of the batch size. Table 2 shows the results. The performance of MMD-GAN training clearly relies heavily on FSR. This penalty not only stabilizes the critic but it can also provides additional learning signal to the generator. Because these penalties are important to the performance of MMD-GANs, it requires tuning several weighting parameters, which need to be set carefully for successful training.

4.4 Effect of the Critic Dimensionality

We examine how changing the dimensionality K of the critic affects the FID of our method. We use the Cifar10 data. Results are shown in Figure 3. Interestingly, we find that there are two regimes: the output dimensionality steadily improves the FID until $K = 1000$, but larger values sharply degrade performance. This agrees with the intuition in Section 3.1 that dimensionality reduction is especially useful for an "intrinsically low dimensional" distribution.

5 Summary

We propose a new criterion for training deep generative networks using the maximum mean discrepancy (MMD) criterion. While MMD networks alone fail to generate high dimensional or color images of good quality, their performance can be greatly improved by training them under a low dimensional mapping. We propose a novel training method for learning this mapping that is based on matching density ratios, which leads to sizeable improvements in performance compared to the recently proposed adversarial methods for training MMD networks.

References

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

Bińkowski, Mikołaj, Sutherland, Dougal J, Arbel, Michael, and Gretton, Arthur. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

Dziugaite, Gintare Karolina, Roy, Daniel M, and Ghahramani, Zoubin. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

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

Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6626–6637. 2017.

Li, Chun-Liang, Chang, Wei-Cheng, Cheng, Yu, Yang, Yiming, and Póczos, Barnabás. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pp. 2200–2210, 2017.

Li, Yujia, Swersky, Kevin, and Zemel, Rich. Generative moment matching networks. In *International Conference on Machine Learning*, pp. 1718–1727, 2015.

Lucic, Mario, Kurach, Karol, Michalski, Marcin, Gelly, Sylvain, and Bousquet, Olivier. Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*, 2017.

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

Sugiyama, Masashi, Yamada, Makoto, von Bünau, Paul, Suzuki, Taiji, Kanamori, Takafumi, and Kawanabe, Motoaki. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks*, 24 2:183–98, 2011.

Sugiyama, Masashi, Suzuki, Taiji, and Kanamori, Takafumi. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.