MMD GAN: Towards Deeper Understanding of Moment Matching Network

Chun-Liang Li[1,*] **Wei-Cheng Chang**[1,*] **Yu Cheng**[2] **Yiming Yang**[1] **Barnabás Póczos**[1]
[1] Carnegie Mellon University, [2]AI Foundations, IBM Research
{chunlial,wchang2,yiming,bapoczos}@cs.cmu.edu chengyu@us.ibm.com
([*] denotes equal contribution)

Abstract

Generative moment matching network (GMMN) is a deep generative model that differs from Generative Adversarial Network (GAN) by replacing the discriminator in GAN with a two-sample test based on kernel maximum mean discrepancy (MMD). Although some theoretical guarantees of MMD have been studied, the empirical performance of GMMN is still not as competitive as that of GAN on challenging and large benchmark datasets. The computational efficiency of GMMN is also less desirable in comparison with GAN, partially due to its requirement for a rather large batch size during the training. In this paper, we propose to improve both the model expressiveness of GMMN and its computational efficiency by introducing *adversarial kernel learning* techniques, as the replacement of a fixed Gaussian kernel in the original GMMN. The new approach combines the key ideas in both GMMN and GAN, hence we name it *MMD GAN*. The new distance measure in MMD GAN is a meaningful loss that enjoys the advantage of weak* topology and can be optimized via gradient descent with relatively small batch sizes. In our evaluation on multiple benchmark datasets, including MNIST, CIFAR-10, CelebA and LSUN, the performance of MMD GAN significantly outperforms GMMN, and is competitive with other representative GAN works.

1 Introduction

The essence of unsupervised learning models the underlying distribution $\mathbb{P}_\mathcal{X}$ of the data $\mathcal{X}$. *Deep generative model* [1, 2] uses deep learning to approximate the distribution of complex datasets with promising results. However, modeling arbitrary density is a statistically challenging task [3]. In many applications, such as caption generation [4], accurate density estimation is not even necessary since we are only interested in *sampling* from the approximated distribution.

Rather than estimating the density of $\mathbb{P}_\mathcal{X}$, Generative Adversarial Network (GAN) [5] starts from a base distribution $\mathbb{P}_\mathcal{Z}$ over $\mathcal{Z}$, such as Gaussian distribution, then trains a transformation network g_θ such that $\mathbb{P}_\theta \approx \mathbb{P}_\mathcal{X}$, where $\mathbb{P}_\theta$ is the underlying distribution of $g_\theta(z)$ and $z \sim \mathbb{P}_\mathcal{Z}$. During the training, GAN-based algorithms require an auxiliary network f_ϕ to estimate the distance between $\mathbb{P}_\mathcal{X}$ and $\mathbb{P}_\theta$. Different probabilistic (pseudo) metrics have been studied [5–8] under GAN framework.

Instead of training an auxiliary network f_ϕ for measuring the distance between $\mathbb{P}_\mathcal{X}$ and $\mathbb{P}_\theta$, Generative moment matching network (GMMN) [9, 10] uses kernel maximum mean discrepancy (MMD) [11], which is the centerpiece of nonparametric two-sample test, to determine the distribution distances. During the training, g_θ is trained to pass the hypothesis test (minimize MMD distance). [11] shows even the simple Gaussian kernel enjoys the strong theoretical guarantees (Theorem 1). However, the empirical performance of GMMN does not meet its theoretical properties. There is no promising empirical results comparable with GAN on challenging benchmarks [12, 13]. Computationally,

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

it also requires larger batch size than GAN needs for training, which is considered to be less efficient [9, 10, 14, 8]

In this work, we try to improve GMMN and consider using MMD with adversarially learned kernels instead of fixed Gaussian kernels to have better hypothesis testing power. The main contributions of this work are:

- In Section 2, we prove that training g_θ via MMD with learned kernels is continuous and differentiable, which guarantees the model can be trained by gradient descent. Second, we prove a new distance measure via kernel learning, which is a sensitive loss function to the distance between $\mathbb{P}_\mathcal{X}$ and $\mathbb{P}_\theta$ (weak* topology). Empirically, the loss decreases when two distributions get closer.
- In Section 3, we propose a practical realization called MMD GAN that learns generator g_θ with the adversarially trained kernel. We further propose a feasible set reduction to speed up and stabilize the training of MMD GAN.
- In Section 5, we show that MMD GAN is computationally more efficient than GMMN, which can be trained with much smaller batch size. We also demonstrate that MMD GAN has promising results on challenging datasets, including CIFAR-10, CelebA and LSUN, where GMMN fails. To our best knowledge, we are the first MMD based work to achieve comparable results with other GAN works on these datasets.

Finally, we also study the connection to existing works in Section 4. Interestingly, we show Wasserstein GAN [8] is the special case of the proposed MMD GAN under certain conditions. The unified view shows more connections between moment matching and GAN, which can potentially inspire new algorithms based on well-developed tools in statistics [15]. Our experiment code is available at https://github.com/OctoberChang/MMD-GAN.

2 GAN, Two-Sample Test and GMMN

Assume we are given data $\{x_i\}_{i=1}^n$, where $x_i \in \mathcal{X}$ and $x_i \sim \mathbb{P}_\mathcal{X}$. If we are interested in sampling from $\mathbb{P}_\mathcal{X}$, it is not necessary to estimate the density of $\mathbb{P}_\mathcal{X}$. Instead, Generative Adversarial Network (GAN) [5] trains a generator g_θ parameterized by θ to transform samples $z \sim \mathbb{P}_\mathcal{Z}$, where $z \in \mathcal{Z}$, into $g_\theta(z) \sim \mathbb{P}_\theta$ such that $\mathbb{P}_\theta \approx \mathbb{P}_\mathcal{X}$. To measure the similarity between $\mathbb{P}_\mathcal{X}$ and $\mathbb{P}_\theta$ via their samples $\{x\}_{i=1}^n$ and $\{g_\theta(z_j)\}_{j=1}^n$ during the training, [5] trains the discriminator f_ϕ parameterized by ϕ for help. The learning is done by playing a two-player game, where f_ϕ tries to distinguish x_i and $g_\theta(z_j)$ while g_θ aims to confuse f_ϕ by generating $g_\theta(z_j)$ similar to x_i.

On the other hand, distinguishing two distributions by finite samples is known as *Two-Sample Test* in statistics. One way to conduct two-sample test is via kernel maximum mean discrepancy (MMD) [11]. Given two distributions $\mathbb{P}$ and $\mathbb{Q}$, and a kernel k, the square of MMD distance is defined as

$$M_k(\mathbb{P}, \mathbb{Q}) = \|\mu_\mathbb{P} - \mu_\mathbb{Q}\|_\mathcal{H}^2 = \mathbb{E}_\mathbb{P}[k(x, x')] - 2\mathbb{E}_{\mathbb{P},\mathbb{Q}}[k(x, y)] + \mathbb{E}_\mathbb{Q}[k(y, y')].$$

Theorem 1. *[11] Given a kernel k, if k is a characteristic kernel, then $M_k(\mathbb{P}, \mathbb{Q}) = 0$ iff $\mathbb{P} = \mathbb{Q}$.*

GMMN: One example of characteristic kernel is Gaussian kernel $k(x, x') = \exp(\|x - x'\|^2)$. Based on Theorem 1, [9, 10] propose generative moment-matching network (GMMN), which trains g_θ by

$$\min_\theta M_k(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta), \tag{1}$$

with a fixed Gaussian kernel k rather than training an additional discriminator f as GAN.

2.1 MMD with Kernel Learning

In practice we use finite samples from distributions to estimate MMD distance. Given $X = \{x_1, \cdots, x_n\} \sim \mathbb{P}$ and $Y = \{y_1, \cdots, y_n\} \sim \mathbb{Q}$, one estimator of $M_k(\mathbb{P}, \mathbb{Q})$ is

$$\hat{M}_k(X, Y) = \frac{1}{\binom{n}{2}} \sum_{i \neq i'} k(x_i, x_i') - \frac{2}{\binom{n}{2}} \sum_{i \neq j} k(x_i, y_j) + \frac{1}{\binom{n}{2}} \sum_{j \neq j'} k(y_j, y_j').$$

Because of the sampling variance, $\hat{M}(X, Y)$ may not be zero even when $\mathbb{P} = \mathbb{Q}$. We then conduct hypothesis test with null hypothesis $H_0 : \mathbb{P} = \mathbb{Q}$. For a given allowable probability of false rejection α,

we can only reject H_0, which imply $\mathbb{P} \neq \mathbb{Q}$, if $\hat{M}(X, Y) > c_\alpha$ for some chose threshold $c_\alpha > 0$. Otherwise, $\mathbb{Q}$ passes the test and $\mathbb{Q}$ is indistinguishable from $\mathbb{P}$ under this test. Please refer to [11] for more details.

Intuitively, if kernel k cannot result in high MMD distance $M_k(\mathbb{P}, \mathbb{Q})$ when $\mathbb{P} \neq \mathbb{Q}$, $\hat{M}_k(\mathbb{P}, \mathbb{Q})$ has more chance to be smaller than c_α. Then we are unlikely to reject the null hypothesis H_0 with finite samples, which implies $\mathbb{Q}$ is not distinguishable from $\mathbb{P}$. Therefore, instead of training g_θ via (1) with a pre-specified kernel k as GMMN, we consider training g_θ via

$$\min_\theta \max_{k \in \mathcal{K}} M_k(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta), \tag{2}$$

which takes different possible characteristic kernels $k \in \mathcal{K}$ into account. On the other hand, we could also view (2) as replacing the fixed kernel k in (1) with the *adversarially learned kernel* $\arg\max_{k \in \mathcal{K}} M_k(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta)$ to have stronger signal where $\mathbb{P} \neq \mathbb{P}_\theta$ to train g_θ. We refer interested readers to [16] for more rigorous discussions about testing power and increasing MMD distances.

However, it is difficult to optimize over all characteristic kernels when we solve (2). By [11, 17] if f is a injective function and k is characteristic, then the resulted kernel $\tilde{k} = k \circ f$, where $\tilde{k}(x, x') = k(f(x), f(x'))$ is still characteristic. If we have a family of injective functions parameterized by ϕ, which is denoted as f_ϕ, we are able to change the objective to be

$$\min_\theta \max_\phi M_{k \circ f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta), \tag{3}$$

In this paper, we consider the case that combining Gaussian kernels with injective functions f_ϕ, where $\tilde{k}(x, x') = \exp(-\|f_\phi(x) - f_\phi(x)'\|^2)$. One example function class of f is $\{f_\phi | f_\phi(x) = \phi x, \phi > 0\}$, which is equivalent to the kernel bandwidth tuning. A more complicated realization will be discussed in Section 3. Next, we abuse the notation $M_{f_\phi}(\mathbb{P}, \mathbb{Q})$ to be MMD distance given the composition kernel of Gaussian kernel and f_ϕ in the following. Note that [18] considers the linear combination of characteristic kernels, which can also be incorporated into the discussed composition kernels. A more general kernel is studied in [19].

2.2 Properties of MMD with Kernel Learning

[8] discuss different distances between distributions adopted by existing deep learning algorithms, and show many of them are discontinuous, such as Jensen-Shannon divergence [5] and Total variation [7], except for Wasserstein distance. The discontinuity makes the gradient descent infeasible for training. From (3), we train g_θ via minimizing $\max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta)$. Next, we show $\max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta)$ also enjoys the advantage of being a continuous and differentiable objective in θ under mild assumptions.

Assumption 2. *$g : \mathcal{Z} \times \mathbb{R}^m \to \mathcal{X}$ is locally Lipschitz, where $\mathcal{Z} \subseteq \mathbb{R}^d$. We will denote $g_\theta(z)$ the evaluation on (z, θ) for convenience. Given f_ϕ and a probability distribution $\mathbb{P}_z$ over $\mathcal{Z}$, g satisfies Assumption 2 if there are local Lipschitz constants $L(\theta, z)$ for $f_\phi \circ g$, which is independent of ϕ, such that $\mathbb{E}_{z \sim \mathbb{P}_z}[L(\theta, z)] < +\infty$.*

Theorem 3. *The generator function g_θ parameterized by θ is under Assumption 2. Let $\mathbb{P}_\mathcal{X}$ be a fixed distribution over $\mathcal{X}$ and Z be a random variable over the space $\mathcal{Z}$. We denote $\mathbb{P}_\theta$ the distribution of $g_\theta(Z)$, then $\max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta)$ is continuous everywhere and differentiable almost everywhere in θ.*

If g_θ is parameterized by a feed-forward neural network, it satisfies Assumption 2 and can be trained via gradient descent as well as propagation, since the objective is continuous and differentiable followed by Theorem 3. More technical discussions are shown in Appendix B.

Theorem 4. *(weak* topology) Let $\{\mathbb{P}_n\}$ be a sequence of distributions. Considering $n \to \infty$, under mild Assumption, $\max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_n) \to 0 \iff \mathbb{P}_n \xrightarrow{D} \mathbb{P}_\mathcal{X}$, where $\xrightarrow{D}$ means* converging in distribution *[3].*

Theorem 4 shows that $\max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_n)$ is a sensible cost function to the distance between $\mathbb{P}_\mathcal{X}$ and $\mathbb{P}_n$. The distance is decreasing when $\mathbb{P}_n$ is getting closer to $\mathbb{P}_\mathcal{X}$, which benefits the supervision of the improvement during the training. All proofs are omitted to Appendix A. In the next section, we introduce a practical realization of training g_θ via optimizing $\min_\theta \max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta)$.

3 MMD GAN

To approximate (3), we use neural networks to parameterized g_θ and f_ϕ with expressive power. For g_θ, the assumption is locally Lipschitz, where commonly used feed-forward neural networks satisfy this constraint. Also, the gradient $\bigtriangledown_\theta \left(\max_\phi f_\phi \circ g_\theta\right)$ has to be bounded, which can be done by clipping ϕ [8] or gradient penalty [20]. The non-trivial part is f_ϕ has to be injective. For an injective function f, there exists an function f^{-1} such that $f^{-1}(f(x)) = x, \forall x \in \mathcal{X}$ and $f^{-1}(f(g(z))) = g(z), \forall z \in \mathcal{Z}$[1], which can be approximated by an autoencoder. In the following, we denote $\phi = \{\phi_e, \phi_d\}$ to be the parameter of discriminator networks, which consists of an encoder f_{ϕ_e}, and train the corresponding decoder $f_{\phi_d} \approx f^{-1}$ to regularize f. The objective (3) is relaxed to be

$$\min_\theta \max_\phi M_{f_{\phi_e}}(\mathbb{P}(\mathcal{X}), \mathbb{P}(g_\theta(\mathcal{Z}))) - \lambda \mathbb{E}_{y \in \mathcal{X} \cup g(\mathcal{Z})} \|y - f_{\phi_d}(f_{\phi_e}(y))\|^2. \tag{4}$$

Note that we ignore the autoencoder objective when we train θ, but we use (4) for a concise presentation. We note that the empirical study suggests autoencoder objective is not necessary to lead the successful GAN training as we will show in Section 5, even though the injective property is required in Theorem 1.

The proposed algorithm is similar to GAN [5], which aims to optimize two neural networks g_θ and f_ϕ in a minmax formulation, while the meaning of the objective is different. In [5], f_{ϕ_e} is a discriminator (binary) classifier to distinguish two distributions. In the proposed algorithm, distinguishing two distribution is still done by two-sample test via MMD, but with an adversarially learned kernel parametrized by f_{ϕ_e}. g_θ is then trained to pass the hypothesis test. More connection and difference with related works is discussed in Section 4. Because of the similarity of GAN, we call the proposed algorithm *MMD GAN*. We present an implementation with the weight clipping in Algorithm 1, but one can easily extend to other Lipschitz approximations, such as gradient penalty [20].

Algorithm 1: MMD GAN, our proposed algorithm.

input : α the learning rate, c the clipping parameter, B the batch size, n_c the number of iterations of discriminator per generator update.

initialize generator parameter θ and discriminator parameter ϕ;

while θ *has not converged* **do**

  for $t = 1, \ldots, n_c$ **do**

    Sample a minibatches $\{x_i\}_{i=1}^B \sim \mathbb{P}(\mathcal{X})$ and $\{z_j\}_{j=1}^B \sim \mathbb{P}(\mathcal{Z})$

    $g_\phi \leftarrow \nabla_\phi M_{f_{\phi_e}}(\mathbb{P}(\mathcal{X}), \mathbb{P}(g_\theta(\mathcal{Z}))) - \lambda \mathbb{E}_{y \in \mathcal{X} \cup g(\mathcal{Z})} \|y - f_{\phi_d}(f_{\phi_e}(y))\|^2$

    $\phi \leftarrow \phi + \alpha \cdot \text{RMSProp}(\phi, g_\phi)$

    $\phi \leftarrow \text{clip}(\phi, -c, c)$

  Sample a minibatches $\{x_i\}_{i=1}^B \sim \mathbb{P}(\mathcal{X})$ and $\{z_j\}_{j=1}^B \sim \mathbb{P}(\mathcal{Z})$

  $g_\theta \leftarrow \nabla_\theta M_{f_{\phi_e}}(\mathbb{P}(\mathcal{X}), \mathbb{P}(g_\theta(\mathcal{Z})))$

  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$

Encoding Perspective of MMD GAN: Besides from using kernel selection to explain MMD GAN, the other way to see the proposed MMD GAN is viewing f_{ϕ_e} as a feature transformation function, and the kernel two-sample test is performed on this transformed feature space (i.e., the code space of the autoencoder). The optimization is finding a manifold with stronger signals for MMD two-sample test. From this perspective, [9] is the special case of MMD GAN if f_{ϕ_e} is the identity mapping function. In such circumstance, the kernel two-sample test is conducted in the original data space.

3.1 Feasible Set Reduction

Theorem 5. *For any f_ϕ, there exists f'_ϕ such that $M_{f_\phi}(\mathbb{P}_r, \mathbb{P}_\theta) = M_{f'_\phi}(\mathbb{P}_r, \mathbb{P}_\theta)$ and $\mathbb{E}_x[f_\phi(x)] \succeq \mathbb{E}_z[f_{\phi'}(g_\theta(z))]$.*

With Theorem 5, we could reduce the feasible set of ϕ during the optimization by solving

$$\min_\theta \max_\phi M_{f_\phi}(\mathbb{P}_r, \mathbb{P}_\theta) \quad s.t.\ \mathbb{E}[f_\phi(x)] \succeq \mathbb{E}[f_\phi(g_\theta(z))]$$

[1] Note that injective is not necessary invertible.

which the optimal solution is still *equivalent* to solving (2).

However, it is hard to solve the constrained optimization problem with backpropagation. We relax the constraint by ordinal regression [21] to be

$$\min_{\theta}\max_{\phi} M_{f_\phi}(\mathbb{P}_r,\mathbb{P}_\theta)+\lambda\min\left(\mathbb{E}[f_\phi(x)]-\mathbb{E}[f_\phi(g_\theta(z))],0\right),$$

which only penalizes the objective when the constraint is violated. In practice, we observe that reducing the feasible set makes the training faster and stabler.

4 Related Works

There has been a recent surge on improving GAN [5]. We review some related works here.

Connection with WGAN: If we composite f_ϕ with linear kernel instead of Gaussian kernel, and restricting the output dimension h to be 1, we then have the objective

$$\min_{\theta}\max_{\phi} \|\mathbb{E}[f_\phi(x)]-\mathbb{E}[f_\phi(g_\theta(z))]\|^2. \tag{5}$$

Parameterizing f_ϕ and g_θ with neural networks and assuming $\exists\phi'\in\Phi$ such $f'_\phi=-f_\phi,\forall\Phi$, recovers Wasserstein GAN (WGAN) [8] [2]. If we treat $f_\phi(x)$ as the data transform function, WGAN can be interpreted as first-order moment matching (linear kernel) while MMD GAN aims to match infinite order of moments with Gaussian kernel form Taylor expansion [9]. Theoretically, Wasserstein distance has similar theoretically guarantee as Theorem 1, 3 and 4. In practice, [22] show neural networks does not have enough capacity to approximate Wasserstein distance. In Section 5, we demonstrate matching high-order moments benefits the results. [23] also propose McGAN that matches second order moment from the primal-dual norm perspective. However, the proposed algorithm requires matrix (tensor) decompositions because of exact moment matching [24], which is hard to scale to higher order moment matching. On the other hand, by giving up exact moment matching, MMD GAN can match high-order moments with kernel tricks. More detailed discussions are in Appendix B.3.

Difference from Other Works with Autoencoders: Energy-based GANs [7, 25] also utilizes the autoencoder (AE) in its discriminator from the energy model perspective, which minimizes the reconstruction error of real samples x while maximize the reconstruction error of generated samples $g_\theta(z)$. In contrast, MMD GAN uses AE to approximate invertible functions by minimizing the reconstruction errors of *both* real samples x and generated samples $g_\theta(z)$. Also, [8] show EBGAN approximates total variation, with the drawback of discontinuity, while MMD GAN optimizes MMD distance. The other line of works [2, 26, 9] aims to match the AE codespace $f(x)$, and utilize the decoder $f_{dec}(\cdot)$. [2, 26] match the distribution of $f(x)$ and z via different distribution distances and generate data (e.g. image) by $f_{dec}(z)$. [9] use MMD to match $f(x)$ and $g(z)$, and generate data via $f_{dec}(g(z))$. The proposed MMD GAN matches the $f(x)$ and $f(g(z))$, and generates data via $g(z)$ directly as GAN. [27] is similar to MMD GAN but it considers KL-divergence without showing continuity and weak* topology guarantee as we prove in Section 2.

Other GAN Works: In addition to the discussed works, there are several extended works of GAN. [28] proposes using the linear kernel to match first moment of its discriminator's latent features. [14] considers the variance of empirical MMD score during the training. Also, [14] only improves the latent feature matching in [28] by using kernel MMD, instead of proposing an adversarial training framework as we studied in Section 2. [29] uses Wasserstein distance to match the distribution of autoencoder loss instead of data. One can consider to extend [29] to higher order matching based on the proposed MMD GAN. A parallel work [30] use energy distance, which can be treated as MMD GAN with different kernel. However, there are some potential problems of its critic. More discussion can be referred to [31].

5 Experiment

We train MMD GAN for image generation on the MNIST [32], CIFAR-10 [33], CelebA [13], and LSUN bedrooms [12] datasets, where the size of training instances are 50K, 50K, 160K, 3M

[2] Theoretically, they are not equivalent but the practical neural network approximation results in the same algorithm.

respectively. All the samples images are generated from a fixed noise random vectors and are not cherry-picked.

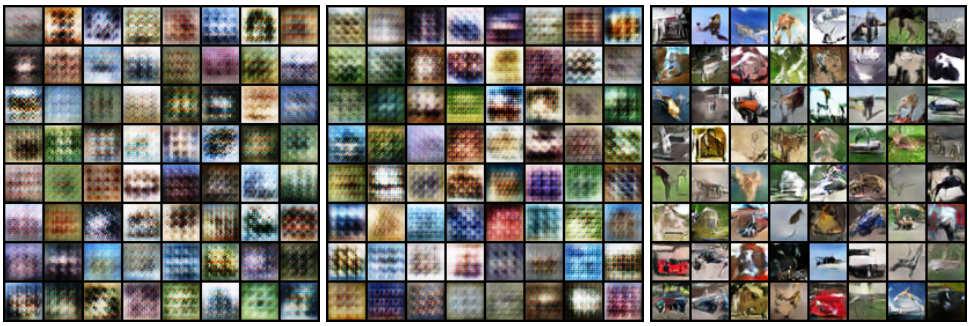
Network architecture: In our experiments, we follow the architecture of DCGAN [34] to design g_θ by its generator and f_ϕ by its discriminator except for expanding the output layer of f_ϕ to be h dimensions.

Kernel designs: The loss function of MMD GAN is implicitly associated with a family of characteristic kernels. Similar to the prior MMD seminal papers [10, 9, 14], we consider a mixture of K RBF kernels $k(x, x') = \sum_{q=1}^{K} k_{\sigma_q}(x, x')$ where k_{σ_q} is a Gaussian kernel with bandwidth parameter σ_q. Tuning kernel bandwidth σ_q optimally still remains an open problem. In this works, we fixed $K = 5$ and σ_q to be $\{1, 2, 4, 8, 16\}$ and left the f_ϕ to learn the kernel (feature representation) under these σ_q.

Hyper-parameters: We use RMSProp [35] with learning rate of 0.00005 for a fair comparison with WGAN as suggested in its original paper [8]. We ensure the boundedness of model parameters of discriminator by clipping the weights point-wisely to the range $[-0.01, 0.01]$ as required by Assumption 2. The dimensionality h of the latent space is manually set according to the complexity of the dataset. We thus use $h = 16$ for MNIST, $h = 64$ for CelebA, and $h = 128$ for CIFAR-10 and LSUN bedrooms. The batch size is set to be $B = 64$ for all datasets.

5.1 Qualitative Analysis

(a) GMMN-D MNIST (b) GMMN-C MNIST (c) MMD GAN MNIST



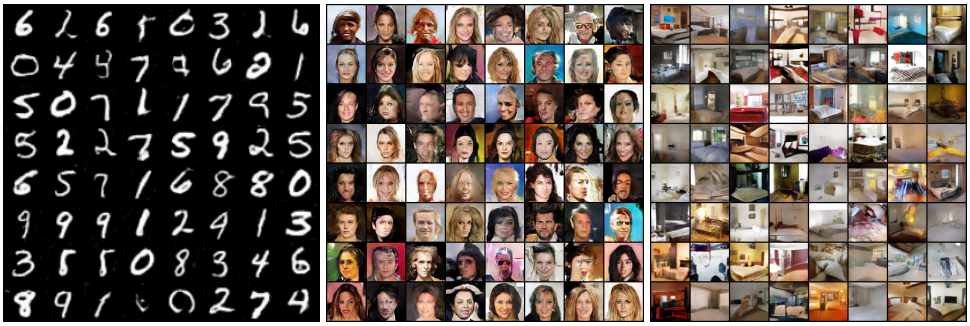
(d) GMMN-D CIFAR-10 (e) GMMN-C CIFAR-10 (f) MMD GAN CIFAR-10

Figure 1: Generated samples from GMMN-D (Dataspace), GMMN-C (Codespace) and our MMD GAN with batch size $B = 64$.

We start with comparing MMD GAN with GMMN on two standard benchmarks, MNIST and CIFAR-10. We consider two variants for GMMN. The first one is original GMMN, which trains the generator by minimizing the MMD distance on the original data space. We call it as *GMMN-D*. To compare with MMD GAN, we also pretrain an autoencoder for projecting data to a manifold, then fix the autoencoder as a feature transformation, and train the generator by minimizing the MMD distance in the code space. We call it as *GMMN-C*.

The results are pictured in Figure 1. Both GMMN-D and GMMN-C are able to generate meaningful digits on MNIST because of the simple data structure. By a closer look, nonetheless, the boundary and shape of the digits in Figure 1a and 1b are often irregular and non-smooth. In contrast, the sample

(a) WGAN MNIST (b) WGAN CelebA (c) WGAN LSUN



(d) MMD GAN MNIST (e) MMD GAN CelebA (f) MMD GAN LSUN

Figure 2: Generated samples from WGAN and MMD GAN on MNIST, CelebA, and LSUN bedroom datasets.

digits in Figure 1c are more natural with smooth outline and sharper strike. For CIFAR-10 dataset, both GMMN variants fail to generate meaningful images, but resulting some low level visual features. We observe similar cases in other complex large-scale datasets such as CelebA and LSUN bedrooms, thus results are omitted. On the other hand, the proposed MMD GAN successfully outputs natural images with sharp boundary and high diversity. The results in Figure 1 confirm the success of the proposed adversarial learned kernels to enrich statistical testing power, which is the key difference between GMMN and MMD GAN.

If we increase the batch size of GMMN to 1024, the image quality is improved, however, it is still not competitive to MMD GAN with $B = 64$. The images are put in Appendix C. This demonstrates that the proposed MMD GAN can be trained more efficiently than GMMN with smaller batch size.

Comparisons with GANs: There are several representative extensions of GANs. We consider recent state-of-art WGAN [8] based on DCGAN structure [34], because of the connection with MMD GAN discussed in Section 4. The results are shown in Figure 2. For MNIST, the digits generated from WGAN in Figure 2a are more unnatural with peculiar strikes. In Contrary, the digits from MMD GAN in Figure 2d enjoy smoother contour. Furthermore, both WGAN and MMD GAN generate diversified digits, avoiding the mode collapse problems appeared in the literature of training GANs. For CelebA, we can see the difference of generated samples from WGAN and MMD GAN. Specifically, we observe varied poses, expressions, genders, skin colors and light exposure in Figure 2b and 2e. By a closer look (view on-screen with zooming in), we observe that faces from WGAN have higher chances to be blurry and twisted while faces from MMD GAN are more spontaneous with sharp and acute outline of faces. As for LSUN dataset, we could not distinguish salient differences between the samples generated from MMD GAN and WGAN.

5.2 Quantitative Analysis

To quantitatively measure the quality and diversity of generated samples, we compute the inception score [28] on CIFAR-10 images. The inception score is used for GANs to measure samples quality and diversity on the pretrained inception model [28]. Models that generate collapsed samples have a relatively low score. Table 1 lists the results for $50K$ samples generated by various unsupervised

generative models trained on CIFAR-10 dataset. The inception scores of [36, 37, 28] are directly derived from the corresponding references.

Although both WGAN and MMD GAN can generate sharp images as we show in Section 5.1, our score is better than other GAN techniques except for DFM [36]. This seems to confirm empirically that higher order of moment matching between the real data and fake sample distribution benefits generating more diversified sample images. Also note DFM appears compatible with our method and combing training techniques in DFM is a possible avenue for future work.

Method	Scores $\pm$ std.
Real data	$11.95 \pm .20$
DFM [36]	7.72
ALI [37]	5.34
Improved GANs [28]	4.36
MMD GAN	**6.17** $\pm$ .07
WGAN	$5.88 \pm .07$
GMMN-C	$3.94 \pm .04$
GMMN-D	$3.47 \pm .03$

Table 1: Inception scores

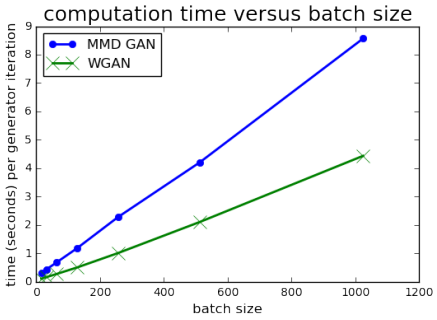
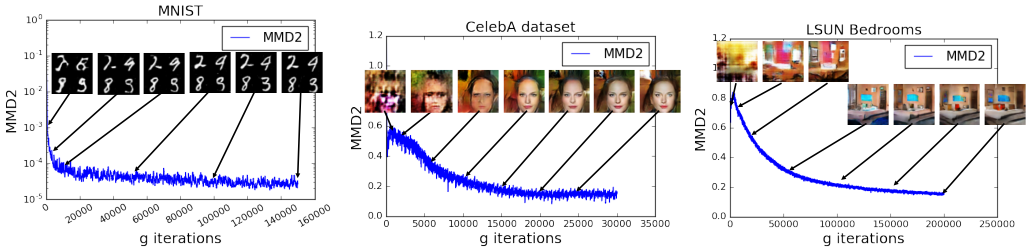


Figure 3: Computation time

5.3 Stability of MMD GAN

We further illustrate how the MMD distance correlates well with the quality of the generated samples. Figure 4 plots the evolution of the MMD GAN estimate the MMD distance during training for MNIST, CelebA and LSUN datasets. We report the average of the $\hat{M}_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta)$ with moving average to smooth the graph to reduce the variance caused by mini-batch stochastic training. We observe during the whole training process, samples generated from the same noise vector across iterations, remain similar in nature. (e.g., face identity and bedroom style are alike while details and backgrounds will evolve.) This qualitative observation indicates valuable stability of the training process. The decreasing curve with the improving quality of images supports the weak* topology shown in Theorem 4. Also, We can see from the plot that the model converges very quickly. In Figure 4b, for example, it converges shortly after tens of thousands of generator iterations on CelebA dataset.



(a) MNIST (b) CelebA (c) LSUN Bedrooms

Figure 4: Training curves and generative samples at different stages of training. We can see a clear correlation between lower distance and better sample quality.

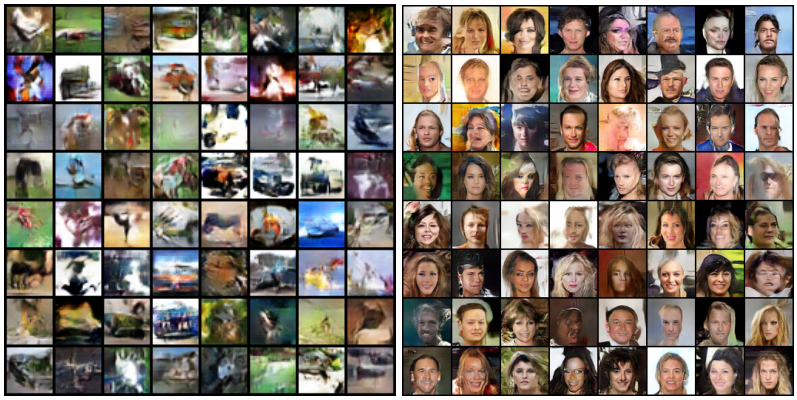
5.4 Computation Issue

We conduct time complexity analysis with respect to the batch size B. The time complexity of each iteration is $O(B)$ for WGAN and $O(KB^2)$ for our proposed MMD GAN with a mixture of K RBF kernels. The quadratic complexity $O(B^2)$ of MMD GAN is introduced by computing kernel matrix, which is sometimes criticized for being inapplicable with large batch size in practice. However, we point that there are several recent works, such as EBGAN [7], also matching pairwise relation between samples of batch size, leading to $O(B^2)$ complexity as well.

Empirically, we find that under GPU environment, the highly parallelized matrix operation tremendously alleviated the quadratic time to almost linear time with modest B. Figure 3 compares the computational time per generator iterations versus different B on Titan X. When $B = 64$, which is adapted for training MMD GAN in our experiments setting, the time per iteration of WGAN and MMD GAN is 0.268 and 0.676 seconds, respectively. When $B = 1024$, which is used for training GMMN in its references [9], the time per iteration becomes 4.431 and 8.565 seconds, respectively. This result coheres our argument that the empirical computational time for MMD GAN is not quadratically expensive compared to WGAN with powerful GPU parallel computation.

5.5 Better Lipschitz Approximation and Necessity of Auto-Encoder

We used weight-clipping for Lipschitz constraint in Assumption 2, another approach for obtaining a discriminator with the similar constraints that approximates a Wasserstein distance is [20], where the gradient of the discriminator is constrained to be 1 between the generated and data points. Inspired by [20], an alternative approach is to apply the gradient constraint as a regularizer to the witness function for a different IPM, such as the MMD. This idea was first proposed in [30] for the Energy Distance (in parallel to our submission), which was shown in [31] to correspond to a gradient penalty on the witness function for any RKHS-based MMD. Here we undertake a preliminary investigation of this approach, where we also drop the requirement in Algorithm 1 for f_ϕ to be injective, which we observe that it is not necessary in practice. We show some preliminary results of training MMD GAN with gradient penalty and without the auto-encoder in Figure 5. The preliminary study indicates that MMD GAN can generate satisfactory results with other Lipschitz constraint approximation. One potential future work is conducting more thorough empirical comparison studies between different approximations.



(a) Cifar10, $Giter = 300K$ (b) CelebA, $Giter = 300K$

Figure 5: MMD GAN results using gradient penalty [20] and without auto-encoder reconstruction loss during training.

6 Discussion

We introduce a new deep generative model trained via MMD with adversarially learned kernels. We further study its theoretical properties and propose a practical realization MMD GAN, which can be trained with much smaller batch size than GMMN and has competitive performances with state-of-the-art GANs. We can view MMD GAN as the first practical step forward connecting moment matching network and GAN. One important direction is applying developed tools in moment matching [15] on general GAN works based the connections shown by MMD GAN. Also, in Section 4, we connect WGAN and MMD GAN by first-order and infinite-order moment matching. [24] shows finite-order moment matching (~ 5) achieves the best performance on domain adaption. One could extend MMD GAN to this by using polynomial kernels. Last, in theory, an injective mapping f_ϕ is *necessary* for the theoretical guarantees. However, we observe that it is not mandatory in practice as we show in Section 5.5. One conjecture is it usually learns the injective mapping with high probability by parameterizing with neural networks, which worth more study as a future work.

Acknowledgments

We thank the reviewers for their helpful comments. We also thank Dougal Sutherland and Arthur Gretton for their valuable feedbacks and pointing out the error in the previous draft. This work is supported in part by the National Science Foundation (NSF) under grants IIS-1546329 and IIS-1563887.

References

[1] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *AISTATS*, 2009.

[2] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2013.

[3] Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[4] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[6] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.

[7] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based Generative Adversarial Network. In *ICLR*, 2017.

[8] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. In *ICML*, 2017.

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

[10] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, 2015.

[11] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *JMLR*, 2012.

[12] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[13] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *CVPR*, 2015.

[14] Dougal J. Sutherland, Hsiao-Yu Fish Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alexander J. Smola, and Arthur Gretton. Generative models and model criticism via optimized maximum mean discrepancy. In *ICLR*, 2017.

[15] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyonds. *arXiv preprint arXiv:1605.09522*, 2016.

[16] Kenji Fukumizu, Arthur Gretton, Gert R Lanckriet, Bernhard Schölkopf, and Bharath K Sriperumbudur. Kernel choice and classifiability for rkhs embeddings of probability distributions. In *NIPS*, 2009.

[17] A. Gretton, B. Sriperumbudur, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, and K. Fukumizu. Optimal kernel choice for large-scale two-sample tests. In *NIPS*, 2012.

[18] Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *NIPS*, 2012.

[19] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *AISTATS*, 2016.

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

[21] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal regression. 1999.

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

[23] Youssef Mroueh, Tom Sercu, and Vaibhava Goel. Mcgan: Mean and covariance feature matching gan. *arxiv pre-print 1702.08398*, 2017.

[24] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017.

[25] Shuangfei Zhai, Yu Cheng, Rogério Schmidt Feris, and Zhongfei Zhang. Generative adversarial networks as variational training of energy based models. *CoRR*, abs/1611.01799, 2016.

[26] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[27] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Adversarial generator-encoder networks. *arXiv preprint arXiv:1704.02304*, 2017.

[28] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016.

[29] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[30] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

[31] Arthur Gretton. Notes on the cramer gan. https://medium.com/towards-data-science/notes-on-the-cramer-gan-752abd505c00, 2017. Accessed: 2017-11-2.

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[33] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[34] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[35] 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*, 2012.

[36] D Warde-Farley and Y Bengio. Improving generative adversarial networks with denoising feature matching. In *ICLR*, 2017.

[37] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. In *ICLR*, 2017.

[38] Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R.G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *JMLR*, 2010.

A Technical Proof

A.1 Proof of Theorem 3

Proof. Since MMD is a probabilistic metric [11], we have the triangular inequality for every M_f. Therefore,

$$\begin{aligned} |\max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta) - \max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_{\theta'})| \leq & \quad \max_\phi |M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta) - M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_{\theta'})| \quad (6) \\ = & \quad |M_{f_{\phi^*}}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta) - M_{f_{\phi^*}}(\mathbb{P}_\mathcal{X}, \mathbb{P}_{\theta'})| \\ \leq & \quad M_{f_{\phi^*}}(\mathbb{P}_\theta, \mathbb{P}_{\theta'}), \quad (7) \end{aligned}$$

where ϕ^* is the solution of (6).

By definition, given any $\phi \in \Phi$, define $h_\theta = f_\phi \circ g_\theta$, the MMD distance $M_{f_\phi}(\mathbb{P}_\theta, \mathbb{P}_{\theta'})$

$$\begin{aligned} = & \quad \mathbb{E}_{z,z'}\left[k(h_\theta(z), h_\theta(z')) - 2k(h_\theta(z), h_{\theta'}(z')) + k(h_{\theta'}(z), h_{\theta'}(z'))\right] \\ \leq & \quad \mathbb{E}_{z,z'}\left[|k(h_\theta(z), h_\theta(z')) - k(h_\theta(z), h_{\theta'}(z'))|\right] + \mathbb{E}_{z,z'}\left[|k(h_{\theta'}(z), h_{\theta'}(z')) - k(h_\theta(z), h_{\theta'}(z'))|\right] \end{aligned} \quad (8)$$

In this, we consider Gaussian kernel k, therefore
$|k(h_\theta(z), h_\theta(z')) - k(h_\theta(z), h_{\theta'}(z'))| = |\exp(-\|h_\theta(z) - h_\theta(z')\|^2) - \exp(-\|h_\theta(z) - h_{\theta'}(z')\|^2)| \leq 1$, for all (θ, θ') and (z, z'). Similarly, $|k(h_{\theta'}(z), h_{\theta'}(z')) - k(h_\theta(z), h_{\theta'}(z'))| \leq 1$. Combining the above claim with (7) and bounded convergence theorem, we have

$$|\max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta) - \max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_{\theta'})| \xrightarrow{\theta \to \theta'} 0,$$

which proves the continuity of $\max_f MMD_f(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta)$.

By Mean Value Theorem, $|e^{-x^2} - e^{-y^2}| \leq \max_z |2ze^{-z^2}| \times |x - y| \leq |x - y|$. Incorporating it with (8) and triangular inequality, we have

$$\begin{aligned} M_{f_\phi}(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \quad \leq & \quad \mathbb{E}_{z,z'}[\|h_\theta(z') - h_{\theta'}(z')\| + \|h_\theta(z) - h_{\theta'}(z)\|] \\ = & \quad 2\mathbb{E}_z[\|h_\theta(z) - h_{\theta'}(z)\|] \end{aligned}$$

Now let h be locally Lipschitz. For a given pair (θ, z) there is a constant $L(\theta, z)$ and an open set U_θ such that for every $(\theta', z) \in U_\theta$ we have

$$\|h_\theta(z) - h_{\theta'}(z)\| \leq L(\theta, z)(\|\theta - \theta'\|)$$

Under Assumption 2, we then achieve

$$|M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta) - M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}'_\theta)| \leq M_{f_\phi}(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \leq 2\mathbb{E}_z[L(\theta, z)]\|\theta - \theta'\|. \quad (9)$$

Combining (7) and (9) implies $\max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta)$ is locally Lipschitz and continuous everywhere. Last, applying Radamacher's theorem proves $\max_\phi M_{f_\phi}(\mathbb{P}_\mathcal{X}, \mathbb{P}_\theta)$ is differentiable almost everywhere, which completes the proof. □

A.2 Proof of Theorem 4

Proof. The proof utilizes parts of results from [38].

($\Rightarrow$) If $\max_\phi M_{f_\phi}(\mathbb{P}_n, \mathbb{P}) \to 0$, there exists $\phi \in \Phi$ such that $M_{f_\phi}(\mathbb{P}, \mathbb{P}_n) \to 0$ since $M_{f_\phi}(\mathbb{P}, \mathbb{P}_n)$ is non-negative. By [38], for any characteristic kernel k,

$$M_k(\mathbb{P}_n, \mathbb{P}) \to 0 \iff \mathbb{P}_n \xrightarrow{D} \mathbb{P}.$$

Therefore, if $\max_\phi M_{f_\phi}(\mathbb{P}_n, \mathbb{P}) \to 0$, $\mathbb{P}_n \xrightarrow{D} \mathbb{P}$.

($\Leftarrow$) By [38], given a characteristic kernel k, if $\sup_{x,x'} k(x, x') \leq 1$, $\sqrt{M_k(\mathbb{P}, \mathbb{Q})} \leq W(\mathbb{P}, \mathbb{Q})$, where $W(\mathbb{P}, \mathbb{Q})$ is Wasserstein distance. In this paper, we consider the kernel $k(x, x') = \exp(-\|f_\phi(x) - f_\phi(x')\|^2) \leq 1$. By above, $\sqrt{M_{f_\phi}(\mathbb{P}, \mathbb{Q})} \leq W(\mathbb{P}, \mathbb{Q}), \forall \phi$. By [8], if $\mathbb{P}_n \xrightarrow{D} \mathbb{P}$, $W(\mathbb{P}, \mathbb{P}_n) \to 0$. Combining all of them, we get

$$\mathbb{P}_n \xrightarrow{D} \mathbb{P} \implies W(\mathbb{P}, \mathbb{P}_n) \to 0 \implies \max_\phi M_{f_\phi}(\mathbb{P}, \mathbb{P}_n) \to 0.$$

□

A.3 Proof of Theorem 5

Proof. The proof assumes $f_\phi(x)$ is scalar, but the vector case can be proved with the same sketch. First, if $\mathbb{E}[f_\phi(x)] > \mathbb{E}[f_\phi(g_\theta(z))]$, then $\phi = \phi'$. If $\mathbb{E}[f_\phi(x)] < \mathbb{E}[f_\phi(g_\theta(z))]$, we let $f = -f_\phi$, then $\mathbb{E}[f(x)] > \mathbb{E}[f(g_\theta(z))]$ and flipping sign does not change the MMD distance. If we parameterized f_ϕ by a neural network, which has a linear output layer, ϕ' can realized by flipping the sign of the weights of the last layer. □

B Property of MMD with Fixed and Learned Kernels

B.1 Continuity and Differentiability

One can simplify Theorem 3 and its proof for standard MMD distance to show MMD is also continuous and differentiable almost everywhere. In [8], they propose a counterexample to show the discontinuity of MMD by assuming $\mathcal{H} = L^2$. However, it is known that L^2 is not in RKHS, so the discussed counterexample is not appropriate.

B.2 IPM Framework

From integral probability metrics (IPM), the probabilistic distance can be defined as

$$\Delta(\mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{F}} \mathbb{E}_{x\sim\mathbb{P}}[f(x)] - \mathbb{E}_{y\sim\mathbb{Q}}[f(y)]. \tag{10}$$

By changing the function class $\mathcal{F}$, we can recover several distances, such as total variation, Wasserstein distance and MMD distance. From [8], the discriminator f_ϕ in different existing works of GAN can be explained to be used to solve different probabilistic metrics based on (10). For MMD, the function class $\mathcal{F}$ is $\{\|f\|_{\mathcal{H}_k} \leq 1\}$, where $\mathcal{H}$ is RKHS associated with kernel k. Different form many distances, such as total variation and Wasserstein distance, there is an analytical representation [11] as we show in Section 2, which is

$$\Delta(\mathbb{P}, \mathbb{Q}) = MMD(\mathbb{P}, \mathbb{Q}) = \|\mu_\mathbb{P} - \mu_\mathbb{Q}\|_\mathcal{H} = \sqrt{\mathbb{E}_\mathbb{P}[k(x, x')] - 2\mathbb{E}_{\mathbb{P},\mathbb{Q}}[k(x, y)] + \mathbb{E}_\mathbb{Q}[k(y, y')]}.$$

Because of the analytical representation of (10), GMMN does not need an additional network f_ϕ for estimating the distance.

Here we also provide an explanation of the proposed MMD with adversarially learned kernel under IPM framework. The MMD distance with adversarially learned kernel is represented as

$$\max_{k\in\mathcal{K}} MMD(\mathbb{P}, \mathbb{Q}),$$

The corresponding IPM formulation is

$$\Delta(\mathbb{P}, \mathbb{Q}) = \max_{k\in\mathcal{K}} MMD(\mathbb{P}, \mathbb{Q}) = \sup_{f\in\mathcal{H}_{k_1}\cup\cdots\cup\mathcal{H}_{k_n}} \mathbb{E}_{x\sim\mathbb{P}}[f(x)] - \mathbb{E}_{y\sim\mathbb{Q}}[f(y)],$$

where $k_i \in \mathcal{K}, \forall i$. From this perspective, the proposed MMD distance with adversarially learned kernel is still defined by IPM but with a larger function class.

B.3 MMD is an Efficient Moment Matching

We consider the example of matching first and second moments of $\mathbb{P}$ and $\mathbb{Q}$. The ℓ_1 objective in McGAN [23] is

$$\|\mu_\mathbb{P} - \mu_\mathbb{Q}\|_1 + \|\Sigma_\mathbb{P} - \Sigma_\mathbb{Q}\|_*,$$

where $\|\cdot\|_*$ is the trace norm. The objective can be changed to be general ℓ_q norm.

In MMD, with polynomial kernel $k(x, y) = 1 + x\top y$, the MMD distance is

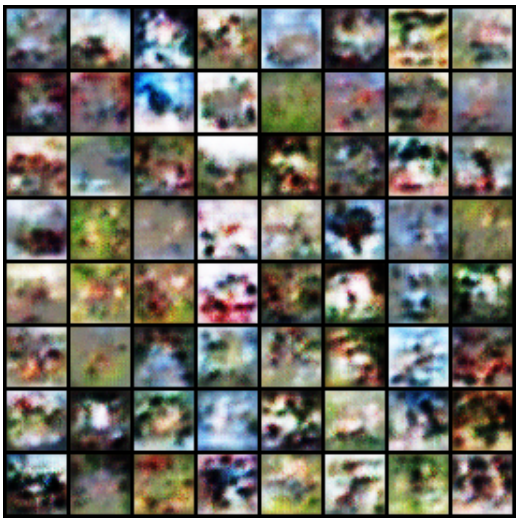
$$2\|\mu_\mathbb{P} - \mu_\mathbb{Q}\|^2 + \|\Sigma_\mathbb{P} - \Sigma_\mathbb{Q} + \mu_\mathbb{P}\mu_\mathbb{P}^\top - \mu_\mathbb{Q}\mu_\mathbb{Q}^\top\|_F^2,$$

which is *inexact* moment matching because the second term contains the quadratic of the first moment. It is difficult to match high-order moments, because we have to deal with high order tensors directly. On the other hand, MMD can easily match high-order moments (even infinite order moments by using Gaussian kernel) with kernel tricks, and enjoys strong theoretical guarantee.

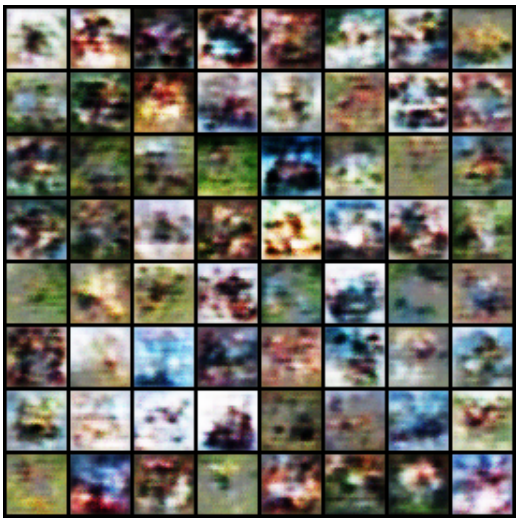
C Performance of GMMN

(a) GMMN-D on MNIST

(b) GMMN-C on MNIST



(c) GMMN-D on CIFAR-10



(d) GMMN-C CIFAR-10

Figure 6: Generative samples from GMMN-D and GMMN-C with training batch size $B = 1024$.