IMPROVING MMD-GAN TRAINING WITH REPULSIVE LOSS FUNCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative adversarial nets (GANs) are widely used to learn the data sampling process and their performance heavily depends on the loss functions used in training. This study revisits MMD-GAN that uses the maximum mean discrepancy (MMD) as the loss function for GAN and makes two contributions. First, we argue that the existing MMD loss function may discourage the learning of data structures as it attempts to attract the discriminator outputs of real data. To address this issue, we propose a repulsive loss function to actively learn the difference among the real data by simply rearranging the terms in MMD. Second, inspired by the hinge loss, we propose a bounded Gaussian kernel to stabilize the training of MMD-GAN. The proposed methods are applied to the unsupervised image generation tasks on CIFAR-10, STL-10, CelebA, and LSUN bedroom datasets. Results show that the repulsive loss function significantly improves over the MMD loss at no additional computational cost and outperforms other representative loss functions. The proposed methods achieved an FID score of 16.21 on the CIFAR-10 dataset using a single DCGAN network and spectral normalization.

1 INTRODUCTION

Generative adversarial nets (GANs) (Goodfellow et al. (2014)) are a branch of generative models that learns to mimic the real data generating process. GANs have been intensively studied in recent years, with a variety of successful applications (Karras et al. (2018); Li et al. (2017b); Lai et al. (2017); Zhu et al. (2017); Ho & Ermon (2016)). The idea of GANs is to jointly train a generator network that attempts to produce artificial samples, and a discriminator network or critic that distinguishes the generated samples from the real ones. Compared to maximum likelihood based methods, GANs tend to produce samples with sharper and more vivid details but they are much harder to train.

Recent studies on improving GAN training have mainly focused on designing loss functions, network architectures and training procedures. A loss function, or simply loss, defines quantitatively the difference of discriminator outputs or scores between real and generated samples. Its gradients are used to train the generator and discriminator. This study focuses on a loss function called maximum mean discrepancy (MMD), which is well known as the distance metric between two probability distributions and widely applied in kernel two-sample test (Gretton et al. (2012)). Theoretically, MMD reaches its global minimum of zero if and only if the tested two distributions are equal. Thus, MMD has been applied to assess the quality of generated samples directly (Li et al. (2015)) and extended to the GAN framework recently (Unterthiner et al. (2018); Li et al. (2017a); Bińkowski et al. (2018)).

In this paper, we interpret the optimization of MMD loss by the discriminator as a combination of attraction and repulsion processes, similar to that of linear discriminant analysis. We argue that the existing MMD loss may discourage the learning of data structures, as the discriminator attempts to minimize the within-group variance of its outputs for the real data. To address this issue, we propose a repulsive loss for the discriminator that explicitly explores the differences among real data. The proposed loss achieved significant improvements over the MMD loss on image generation tasks of four benchmark datasets, without incurring any additional computational cost. Furthermore, a bounded Gaussian kernel is proposed such that using a single kernel in MMD is sufficient, in contrast

to a linear combination of kernels used in Li et al. (2017a) and Bińkowski et al. (2018). This may further bring down the computational cost of the MMD loss in a variety of applications.

The paper is organized as follows. Section 2 reviews the GAN trained using the MMD loss (MMD-GAN). We propose the repulsive loss in Section 3, two practical techniques to improve the model stability in Section 4, and the results of extensive experiments in Section 5. At last, we discuss the connections of our model to existing work in Section 6.

2 MMD-GAN

This section introduces GAN and the MMD loss. Consider a random variable $\mathbf{X} \in \mathcal{X}$ with an empirical data distribution $P_{\mathbf{X}}$ to be learned. A typical GAN model consists of two neural networks: a generator G and a discriminator D. The G maps a latent code $\boldsymbol{z}$ with a fixed distribution $P_{\mathbf{Z}}$, e.g., Gaussian, to the data space $\mathcal{X}$: $\boldsymbol{y} = G(\boldsymbol{z}) \in \mathcal{X}$, where $\boldsymbol{y}$ is called the generated samples with distribution P_G. The D maps a sample $\boldsymbol{a}$ to some score $D(\boldsymbol{a}) \in \mathbb{R}^d$, where $\boldsymbol{a}$ may be $\boldsymbol{y}$ or a real sample $\boldsymbol{x}$. We assume linear activation is used at the last layer of D.

Several loss functions have been proposed to calculate the difference between the scores of real and generated samples: $\{D(\boldsymbol{x})\}$ and $\{D(\boldsymbol{y})\}$, including

- The Minimax loss (Goodfellow et al. (2014)): $L_D = \mathbb{E}_{P_{\mathbf{X}}}[\text{Softplus}(-D(\boldsymbol{x}))] + \mathbb{E}_{P_{\mathbf{Z}}}[\text{Softplus}(D(G(\boldsymbol{z})))]$ and $L_G = -L_D$, which can be derived from the Jensen–Shannon (JS) divergence between $P_{\mathbf{X}}$ and the model distribution P_G.
- The non-saturating loss (Goodfellow et al. (2014)) is a variant of the minimax loss with the same L_D and $L_G = \mathbb{E}_{P_{\mathbf{Z}}}[\text{Softplus}(-D(G(\boldsymbol{z})))]$.
- The Hinge loss (Tran et al. (2017)): $L_D = \mathbb{E}_{P_{\mathbf{X}}}[\text{ReLU}(1 - D(\boldsymbol{x}))] + \mathbb{E}_{P_{\mathbf{Z}}}[\text{ReLU}(1 + D(G(\boldsymbol{z})))]$, $L_G = \mathbb{E}_{P_{\mathbf{Z}}}[-D(G(\boldsymbol{z}))]$, which is notably known for usage in support vector machines and is related to the total variation (TV) distance (Nguyen et al. (2009)).
- The Wasserstein loss (Arjovsky et al. (2017); Gulrajani et al. (2017)) is derived from the Wasserstein distance between $P_{\mathbf{X}}$ and P_G: $L_G = -\mathbb{E}_{P_{\mathbf{Z}}}[D(G(\boldsymbol{z}))]$, $L_D = \mathbb{E}_{P_{\mathbf{Z}}}[D(G(\boldsymbol{z}))] - \mathbb{E}_{P_{\mathbf{X}}}[D(\boldsymbol{x})]$, where D is subject to some Lipschitz constraint.
- The maximum mean discrepancy (MMD) (Li et al. (2017a); Bińkowski et al. (2018)).

Among them, MMD uses kernel embedding $\boldsymbol{\phi}(\boldsymbol{a}) = k(\cdot, \boldsymbol{a})$ associated with a characteristic kernel k such that $\boldsymbol{\phi}$ is infinite-dimensional and $\langle \boldsymbol{\phi}(\boldsymbol{a}), \boldsymbol{\phi}(\boldsymbol{b}) \rangle_{\mathcal{H}} = k(\boldsymbol{a}, \boldsymbol{b})$. The squared MMD distance between two distributions P and Q is then

$$M_k^2(P, Q) = \|\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q\|_{\mathcal{H}}^2 = \mathbb{E}_{\boldsymbol{a}, \boldsymbol{a}' \sim P}[k(\boldsymbol{a}, \boldsymbol{a}')] + \mathbb{E}_{\boldsymbol{b}, \boldsymbol{b}' \sim Q}[k(\boldsymbol{b}, \boldsymbol{b}')] - 2\mathbb{E}_{\boldsymbol{a} \sim P, \boldsymbol{b} \sim Q}[k(\boldsymbol{a}, \boldsymbol{b})] \quad (1)$$

The kernel $k(\boldsymbol{a}, \boldsymbol{b})$ measures the similarity between two samples $\boldsymbol{a}$ and $\boldsymbol{b}$. Gretton et al. (2012) proved that, using a characteristic kernel k, $M_k^2(P, Q) \geq 0$ with equality applies if and only if $P = Q$.

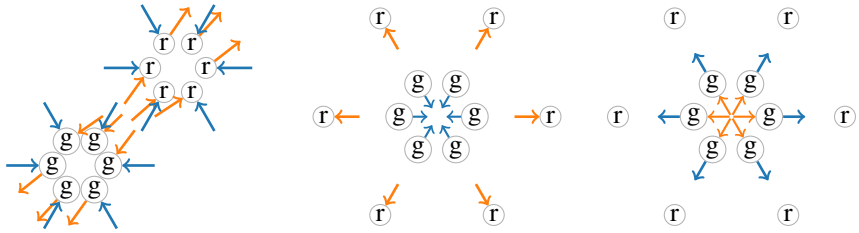
In MMD-GAN, the discriminator D can be interpreted as forming a new kernel with k: $k \circ D(\boldsymbol{a}, \boldsymbol{b}) = k(D(\boldsymbol{a}), D(\boldsymbol{b})) = k_D(\boldsymbol{a}, \boldsymbol{b})$. If D is injective, $k \circ D$ is characteristic and $M_{k \circ D}^2(P_{\mathbf{X}}, P_G)$ reaches its minimum only when $P_{\mathbf{X}} = P_G$ (Li et al. (2017a)). Thus, the objective functions for D and G could be (Li et al. (2017a); Bińkowski et al. (2018)):

$$\min_D L_D^{\text{att}} = -M_{k \circ D}^2(P_{\mathbf{X}}, P_G) = 2\mathbb{E}_{P_{\mathbf{X}}, P_G}[k_D(\boldsymbol{x}, \boldsymbol{y})] - \mathbb{E}_{P_{\mathbf{X}}}[k_D(\boldsymbol{x}, \boldsymbol{x}')] - \mathbb{E}_{P_G}[k_D(\boldsymbol{y}, \boldsymbol{y}')] \quad (2)$$

$$\min_G L_G^{\text{mmd}} = M_{k \circ D}^2(P_{\mathbf{X}}, P_G) = \mathbb{E}_{P_G}[k_D(\boldsymbol{y}, \boldsymbol{y}')] - 2\mathbb{E}_{P_{\mathbf{X}}, P_G}[k_D(\boldsymbol{x}, \boldsymbol{y})] + C \quad (3)$$

where $C = \mathbb{E}_{P_{\mathbf{X}}}[k_D(\boldsymbol{x}, \boldsymbol{x}')]$ is independent of G. This approach has been shown to improve over directly using MMD as the loss function for the generator G (Li et al. (2017a)).

Liu et al. (2017) showed that Wasserstein metric and MMD are weaker objective functions for GAN than the JS-divergence and TV distance, in the sense that convergence of P_G to $P_{\mathbf{X}}$ in JS-divergence and TV distance also implies convergence in Wasserstein metric and MMD. Weak metrics are desirable as they provide more information on adjusting the model to fit the data distribution (Liu et al. (2017)). Nagarajan & Kolter (2017) proved that the GAN trained using the minimax loss and



(a) L_D^{att} (Eq. 2) (b) $L_{D,1}^{\text{rep}}$ (Eq. 5) (c) L_G^{mmd} paired with $L_{D,1}^{\text{rep}}$

Figure 1: Illustration of the gradient directions of each loss on the real sample scores $\{D(\boldsymbol{x})\}$ ("r" nodes) and generated sample scores $\{D(\boldsymbol{y})\}$ ("g" nodes). The blue arrows stand for attraction and the orange arrows for repulsion. When L_G^{mmd} is paired with L_D^{att}, the gradient directions of L_G^{mmd} on $\{D(\boldsymbol{y})\}$ can be obtained by reversing the arrows in (a), thus are omitted.

gradient updates on model parameters is locally exponentially stable near equilibrium, while the GAN using Wasserstein loss is not. In Appendix A, we demonstrate that the MMD-GAN trained using gradient methods is locally exponentially stable near equilibrium.

3 Repulsive Loss Function

In this section, we interpret the MMD-GAN training using L_D^{att} and L_G^{mmd}, both as a combination of attraction and repulsion processes, and propose a novel loss function for the discriminator by rearranging these components.

First, consider a linear discriminant analysis (LDA) model as the discriminator. The task is to find a projection $\boldsymbol{w}$ to maximize the between-group variance $\left\|\boldsymbol{w}^T\boldsymbol{\mu}_x - \boldsymbol{w}^T\boldsymbol{\mu}_y\right\|$ and minimize the within-group variance $\boldsymbol{w}^T(\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y)\boldsymbol{w}$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are group mean and covariance.

We then consider the neural network discriminator in MMD-GAN. By minimizing L_D^{att}, the discriminator D tackles two tasks: 1) D reduces $\mathbb{E}_{P_{\mathbf{X}},P_G}[k_D(\boldsymbol{x},\boldsymbol{y})]$, i.e., repulses the two groups $\{D(\boldsymbol{x})\}$ and $\{D(\boldsymbol{y})\}$ from each other (see Fig 1a orange arrows); and 2) D increases $\mathbb{E}_{P_{\mathbf{X}}}[k_D(\boldsymbol{x},\boldsymbol{x}')]$ and $\mathbb{E}_{P_G}[k(\boldsymbol{y},\boldsymbol{y}')]$, i.e. attracts $\{D(\boldsymbol{x})\}$ and $\{D(\boldsymbol{y})\}$ within each group (see Fig 1a blue arrows). Thus, similar to the LDA method, D may be interpreted as finding nonlinear projections to maximize the between-group variance and minimize the within-group variance of projected scores. We refer to loss functions that attract the real data scores $\{D(\boldsymbol{x})\}$ as attractive losses.

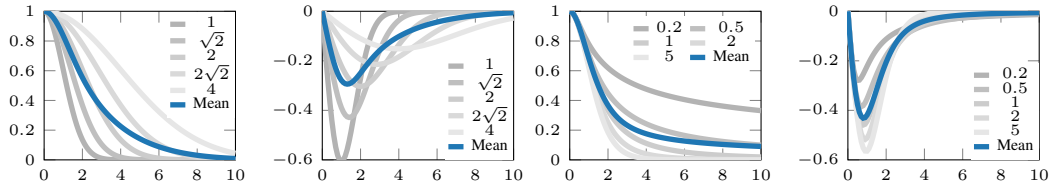
We argue that the attractive loss in Eq. 2 has two issues that may slow down the GAN training:

1. D may focus more on the similarities among real samples (in order to attract $\{D(\boldsymbol{x})\}$) than the fine details that separate them. Initially, G produces low-quality samples and it may be adequate for D to learn the common features of $\{\boldsymbol{x}\}$ in order to distinguish between $\{\boldsymbol{x}\}$ and $\{\boldsymbol{y}\}$. Only when $\{D(\boldsymbol{y})\}$ is sufficiently close to $\{D(\boldsymbol{x})\}$ will D learn the fine details of $\{\boldsymbol{x}\}$ to be able to separate $\{D(\boldsymbol{x})\}$ from $\{D(\boldsymbol{y})\}$. Consequently, D may leave out the data information for G to access during training.

2. As shown in Fig 1a, the gradients on $D(\boldsymbol{y})$ from the attraction (blue arrows) and repulsion (orange arrows) terms in L_D^{att} (and thus L_G^{mmd}) may have opposite directions during training. Their summation may be small in magnitude even when $D(\boldsymbol{y})$ is far away from $D(\boldsymbol{x})$, which may cause G to get stagnated locally.

Therefore, we propose a different tactic called repulsive loss for the discriminator D:

$$L_{D,\lambda}^{\text{rep}} = \lambda\mathbb{E}_{P_{\mathbf{X}}}[k_D(\boldsymbol{x},\boldsymbol{x}')] - (\lambda-1)\mathbb{E}_{P_{\mathbf{X}},P_G}[k_D(\boldsymbol{x},\boldsymbol{y})] - \mathbb{E}_{P_G}[k_D(\boldsymbol{y},\boldsymbol{y}')] \tag{4}$$



(a) (b) (c) (d)

Figure 2: (a) Gaussian kernels $\{k^{\text{rbf}}_{\sigma_i}(\boldsymbol{a}, \boldsymbol{b})\}$ and their mean as a function of $e = \|\boldsymbol{a} - \boldsymbol{b}\|$, where $\sigma_i \in \{1, \sqrt{2}, 2, 2\sqrt{2}, 4\}$ were used in our experiments; (b) derivatives of $\{k^{\text{rbf}}_{\sigma_i}(\boldsymbol{a}, \boldsymbol{b})\}$ in (a); (c) rational quadratic kernel $\{k^{\text{rq}}_{\alpha_i}(\boldsymbol{a}, \boldsymbol{b})\}$ and their mean, where $\alpha_i \in \{0.2, 0.5, 1, 2, 5\}$; (d) derivatives of $\{k^{\text{rq}}_{\alpha_i}(\boldsymbol{a}, \boldsymbol{b})\}$ in (c).

where the weight $\lambda \geq 0$ is a hyper-parameter[1]. In its simplest form ($\lambda = 1$),

$$L^{\text{rep}}_{D,1} = \mathbb{E}_{P_{\mathbf{x}}}[k_D(\boldsymbol{x}, \boldsymbol{x}')] - \mathbb{E}_{P_G}[k_D(\boldsymbol{y}, \boldsymbol{y}')] \tag{5}$$

The generator still uses the same MMD loss as before (see Eq. 3). Thus, the adversary lies in D attracting $\{D(\boldsymbol{y})\}$ via maximizing $\mathbb{E}_{P_G}[k_D(\boldsymbol{y}, \boldsymbol{y}')]$ (see Fig 1b) and G repulsing $\{D(\boldsymbol{y})\}$ (see Fig 1c). On the other hand, D also learns to separate the real data by minimizing $\mathbb{E}_{P_{\mathbf{x}}}[k_D(\boldsymbol{x}, \boldsymbol{x}')]$, which actively explores the fine details in data and may result in more meaningful gradients for G.

Note that in Eq. 5, D does not explicitly push the average score of $\{D(\boldsymbol{y})\}$ away from that of $\{D(\boldsymbol{x})\}$ because it may have no effect on the pair-wise sample distances. But G explicitly matches the average scores. Thus, we believe it is less likely for L^{mmd}_G to yield opposite gradients when paired with $L^{\text{rep}}_{D,1}$ than L^{att}_D (see Fig 1c). At last, we note that the proposed $L^{\text{rep}}_{D,\lambda}$ (Eq. 4) has exactly the same computational cost as $L^{\text{att}}_{D,\lambda}$ (Eq. 2), given that we only rearranged the terms in $L^{\text{att}}_{D,\lambda}$.

4 Regularization on MMD and Discriminator

This section presents two approaches to improve the training of MMD-GAN using L^{att}_D (Eq. 2) as well as the proposed repulsive loss $L^{\text{rep}}_{D,\lambda}$ (Eq. 4). We focus on image generation tasks using convolutional neural networks (CNN) for both G and D. First, we propose a bounded kernel for MMD that improve the model stability; then a generalized power iteration method to estimate the spectral norm of a convolutional kernel, which was used in spectral normalization on the discriminator in all experiments.

4.1 Kernel in MMD

For MMD-GAN, the following two kernels have been used:

- Gaussian radial basis function (RBF), or Gaussian kernel (Li et al. (2017a)), $k^{\text{rbf}}_{\sigma}(\boldsymbol{a}, \boldsymbol{b}) = \exp(-\frac{1}{2\sigma^2}\|\boldsymbol{a} - \boldsymbol{b}\|^2)$ where $\sigma > 0$ is the kernel scale or bandwidth.
- Rational quadratic kernel (Bińkowski et al. (2018)), $k^{\text{rq}}_{\alpha}(\boldsymbol{a}, \boldsymbol{b}) = (1 + \frac{1}{2\alpha}\|\boldsymbol{a} - \boldsymbol{b}\|^2)^{-\alpha}$, where the kernel scale $\alpha > 0$ corresponds to a mixture of Gaussian kernels with a Gamma$(\alpha, 1)$ prior on the inverse kernel scales σ^{-1}.

It is interesting that both studies used a linear combination of kernels with five different kernel scales, i.e., $k_{\text{rbf}} = \sum_{i=1}^{5} k^{\text{rbf}}_{\sigma_i}$ and $k_{\text{rq}} = \sum_{i=1}^{5} k^{\text{rq}}_{\alpha_i}$, where $\sigma_i \in \{1, 2, 4, 8, 16\}$, $\alpha_i \in \{0.2, 0.5, 1, 2, 5\}$ (see Fig 2a and 2c for illustration). We suspect the reason is that a single kernel $k(\boldsymbol{a}, \boldsymbol{b})$ is saturated when the distance $\|\boldsymbol{a} - \boldsymbol{b}\|$ is either too large or too small compared to the kernel scale (see Fig 2b and 2d), which may cause diminishing gradients during training. Both Li et al. (2017a) and Bińkowski et al. (2018) applied penalties on the discriminator parameters but not to the MMD loss. Thus the

[1]The weights for the three terms in $L^{\text{rep}}_{D,\lambda}$ sum up to zero. This is to make sure the $\partial L^{\text{rep}}_{D,\lambda}/\partial \boldsymbol{\theta}_D$ is zero at equilibrium $P_{\mathbf{x}} = P_G$, where $\boldsymbol{\theta}_D$ is the parameters of D.

saturation issue may still exist. Using a linear combination of kernels with different kernel scales may alleviate this issue but not eradicate it.

Inspired by the hinge loss, we propose a bounded RBF (RBF-B) kernel for the discriminator. The idea is to prevent D from pushing $\{D(\boldsymbol{x})\}$ too far away from $\{D(\boldsymbol{y})\}$ and causing saturation. For L_D^{att} in Eq. 2, the RBF-B kernel is:

$$k_\sigma^{\text{rbf-b}}(\boldsymbol{a},\boldsymbol{b}) = \begin{cases} \exp(-\frac{1}{2\sigma^2}\max(\|\boldsymbol{a}-\boldsymbol{b}\|^2, b_l)) & \text{if } \boldsymbol{a},\boldsymbol{b} \in \{D(\boldsymbol{x})\} \text{ or } \boldsymbol{a},\boldsymbol{b} \in \{D(\boldsymbol{y})\} \\ \exp(-\frac{1}{2\sigma^2}\min(\|\boldsymbol{a}-\boldsymbol{b}\|^2, b_u)) & \text{if } \boldsymbol{a} \in \{D(\boldsymbol{x})\} \text{ and } \boldsymbol{b} \in \{D(\boldsymbol{y})\} \end{cases} \tag{6}$$

For $L_{D,\lambda}^{\text{rep}}$ in Eq. 4 and $L_{D,1}^{\text{rep}}$ Eq. 5, the RBF-B kernel is:

$$k_\sigma^{\text{rbf-b}}(\boldsymbol{a},\boldsymbol{b}) = \begin{cases} \exp(-\frac{1}{2\sigma^2}\max(\|\boldsymbol{a}-\boldsymbol{b}\|^2, b_l)) & \text{if } \boldsymbol{a},\boldsymbol{b} \in \{D(\boldsymbol{y})\} \\ \exp(-\frac{1}{2\sigma^2}\min(\|\boldsymbol{a}-\boldsymbol{b}\|^2, b_u)) & \text{if } \boldsymbol{a},\boldsymbol{b} \in \{D(\boldsymbol{x})\} \end{cases} \tag{7}$$

where b_l and b_u are the lower and upper bounds. As such, a single kernel is sufficient and we set $\sigma = 1$, $b_l = 0.25$ and $b_u = 4$ in all experiments for simplicity and leave their tuning for future work.

It should be noted that RBF-B kernel is among many techniques to address the saturation issue. We found other methods, e.g., random sampling kernel scale, instance noise (Sønderby et al. (2017)) and label smoothing Szegedy et al. (2016); Salimans et al. (2016), may also improve the model performance and stability. However, the computational cost of RBF-B kernel is relatively low.

4.2 Spectral Normalization in Discriminator

Arjovsky et al. (2017) showed that, if the supports of $P_{\mathbf{X}}$ and $P_{\boldsymbol{\theta}_G}$ do not overlap, there exists a perfect discriminator to separate $\{\boldsymbol{x}\}$ and $\{\boldsymbol{y}\}$. Without any Lipschitz constraints on D, it may simply increase the magnitude of its outputs to minimize the discriminator loss, causing unstable training. Spectral normalization divides the weight matrix of each layer by its spectral norm, which imposes an upper bound on the magnitudes of outputs and gradients at each layer of D (Miyato et al. (2018)). However, to estimate the spectral norm of a convolution kernel, Miyato et al. (2018) reshaped the kernel into a matrix. We propose a generalized power iteration method to directly estimate the spectral norm of a convolution kernel (see Appendix B.1 for details) and applied spectral normalization to the discriminator in all experiments.

5 Experiments

In this section, we empirically evaluate the proposed 1) repulsive loss $L_{D,\lambda}^{\text{rep}}$ (Section 3) on unsupervised training of GAN for image generation tasks; and 2) RBF-B kernel to stabilize MMD-GAN training. To show the efficacy of $L_{D,\lambda}^{\text{rep}}$, we compared the loss functions $(L_{D,\lambda}^{\text{rep}}, L_G^{\text{mmd}})$ using Gaussian kernel (MMD-rep) with $(L_{D,\lambda}^{\text{att}}, L_G^{\text{mmd}})$ using Gaussian kernel (MMD-rbf) (Li et al. (2017a)) and rational quadratic kernel (MMD-rq) (Bińkowski et al. (2018)), as well as non-saturating loss (Goodfellow et al. (2014)) and hinge loss (Tran et al. (2017)). To show the efficacy of RBF-B kernel, we applied it to both $L_{D,\lambda}^{\text{att}}$ and $L_{D,\lambda}^{\text{rep}}$, resulting in two methods MMD-rbf-b and MMD-rep-b. The Wasserstein loss was excluded for comparison because it cannot be directly used with spectral normalization (Miyato et al. (2018)).

5.1 Experiment Setup

Dataset: The loss functions were evaluated on four datasets: 1) CIFAR-10 ($50K$ images, 32×32 pixels) (Krizhevsky & Hinton (2009)); 2) STL-10 ($100K$ images, 48×48 pixels) (Coates et al. (2011)); 3) CelebA (about $203K$ images, 64×64 pixels) (Liu et al. (2015)); and 4) LSUN bedrooms (around 3 million images, 64×64 pixels) (Yu et al. (2015)). The images were scaled to range $[-1, 1]$ to avoid numeric issues.

Network architecture: The DCGAN (Radford et al. (2016)) architecture from Miyato et al. (2018) was used (see Appendix B.2 for details). In all experiments, batch normalization (BN) (Ioffe & Szegedy (2015)) was used in the generator and spectral normalization with the generalized power

Table 1: Inception score (IS), Fréchet Inception distance (FID) and multi-scale structural similarity (MS-SSIM) on image generation tasks

Methods	CIFAR-10[1]		STL-10[1]		CelebA[1]		LSUN-bedrom[1]	
	IS	FID	IS	FID	FID	MS-SSIM	FID	MS-SSIM
Real data	11.31	2.09	26.37	2.10	1.09	0.2678	1.24	0.0915
Non-saturating	7.39	23.23	8.25	48.53	10.64	0.2895	23.66	0.1027
Hinge	7.33	23.46	8.24	49.44	8.60	0.2894	16.73	0.0946
MMD-rbf[2]	7.05	28.38	8.13	57.52	13.03	0.2937		
MMD-rq[2]	7.22	27.00	8.11	54.05	12.74	0.2935		
MMD-rbf-b	7.18	25.25	8.07	51.86	10.09	0.3090	32.29	0.1001
MMD-rep	7.99	**16.65**	**9.36**	**36.67**	**7.20**	0.2761	16.91	**0.0901**
MMD-rep-b	**8.29**	**16.21**	**9.34**	37.63	**6.79**	**0.2659**	**12.52**	**0.0908**

[1] MS-SSIM is not compatible with CIFAR-10 and STL-10 which have data from many classes; for CelebA and LSUN-bedroom, IS is not meaningful (Bińkowski et al. (2018)); thus they are not shown.
[2] On LSUN-bedroom, MMD-rbf and MMD-rq did not achieve reasonable results and thus are omitted.

iteration (see Appendix B.1) in the discriminator. For MMD related losses, the dimension of discriminator output layer is set to 16; for non-saturating loss and hinge loss, it is set to 1.

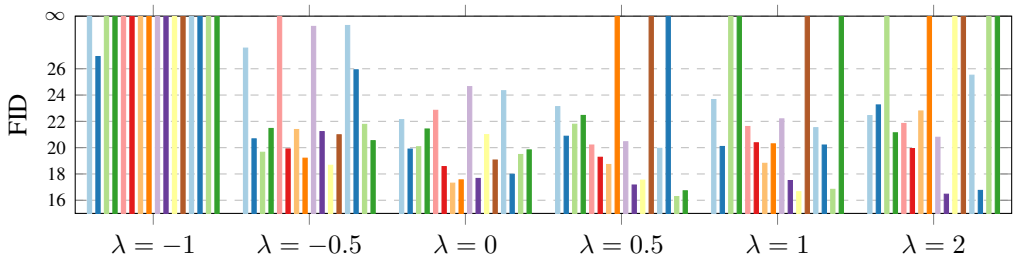
Hyper-parameters: We used Adam optimizer (Kingma & Ba (2015)) with momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$; two-timescale update rule (TTUR) (Heusel et al. (2017)) with two learning rates (ρ_D, ρ_G) chosen from $\{1e\text{-}4, 2e\text{-}4, 5e\text{-}4, 1e\text{-}3\}$ (16 combinations in total); and batch size 64. Fine-tuning on learning rates may improve the model performance, but constant learning rates were used for fair comparison. All models were trained for $100K$ iterations on CIFAR-10, STL-10, CelebA and LSUN bedrooms dataset, with $n_{dis} = 1$, i.e., one discriminator update per generator update[2]. For MMD-rbf, the kernel scales $\sigma_i \in \{1, \sqrt{2}, 2, 2\sqrt{2}, 4\}$ were used due to better performance than the original values in Li et al. (2017a). For MMD-rq, $\alpha_i \in \{0.2, 0.5, 1, 2, 5\}$. For MMD-rbf-b, MMD-rep, MMD-rep-b, a single Gaussian kernel with $\sigma = 1$ was used.

Evaluation metrics: Inception score (IS) (Salimans et al. (2016)), Fréchet Inception distance (FID) (Heusel et al. (2017)) and multi-scale structural similarity (MS-SSIM) (Wang et al. (2003)) were used for quantitative evaluation. Both IS and FID are calculated using a pre-trained Inception model (Szegedy et al. (2016)). Higher IS and lower FID scores indicate better image quality. MS-SSIM calculates the pair-wise image similarity and is used to detect mode collapses among images of the same class (Odena et al. (2017)). Lower MS-SSIM values indicate perceptually more diverse images. For each model, $50K$ randomly generated samples and $50K$ real samples were used to calculate IS, FID and MS-SSIM.

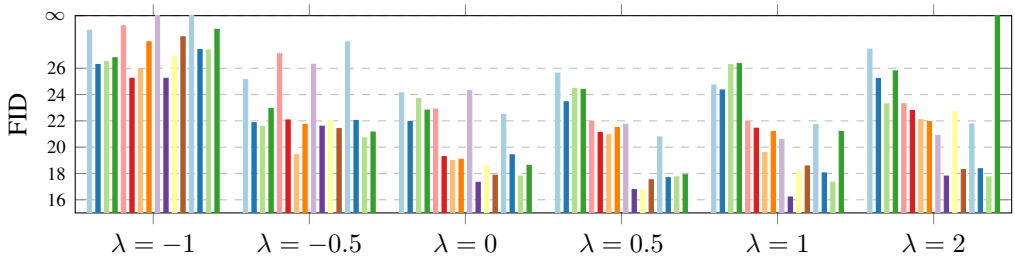
5.2 Quantitative Analysis

Table 1 shows the IS, FID and MS-SSIM scores of applying different loss functions on the benchmark datasets with the optimal learning rate settings tested experimentally. The proposed MMD-rep and MMD-rep-b methods (both using $L_{D,1}^{rep}$ in Eq. 5) achieved the best scores on almost all tasks. We observed that: 1) MMD-rep-b performed significantly better than MMD-rbf-b by changing only the loss function for discriminator, showing the proposed repulsive loss can greatly improve MMD-GAN training; 2) Using a single kernel, MMD-rbf-b still performed better than MMD-rbf and MMD-rq which used a linear combination of five kernels, indicating that the kernel saturation may be an issue that slows down MMD-GAN training; 3) MMD-rep-b performed better than MMD-rep on the CelebA and LSUN-bedroom datasets where we found MMD-rep was not stable on many learning rate combinations.

[2]Increasing or decreasing n_{dis} may improve the model performance in some cases, but it has significant impact on the computation cost. For simple and fair comparison, we set $n_{dis} = 1$ in all cases.



(a) MMD-rep



(b) MMD-rep-b

Figure 3: FID scores of (a) MMD-rep and (b) MMD-rep-b on CIFAR-10 dataset for 16 learning rate combinations and $\lambda \in \{$-1, -0.5, $0, 0.5, 1, 2\}$. For each λ, the order of learning rate combination (ρ_D, ρ_G) is $(1e\text{-}4, 1e\text{-}4)$, $(1e\text{-}4, 2e\text{-}4)$,...,$(1e\text{-}3, 1e\text{-}3)$. We use the infinite ($\infty$) FID score to indicate that the model diverged or produced poor results (FID$>$ 30).

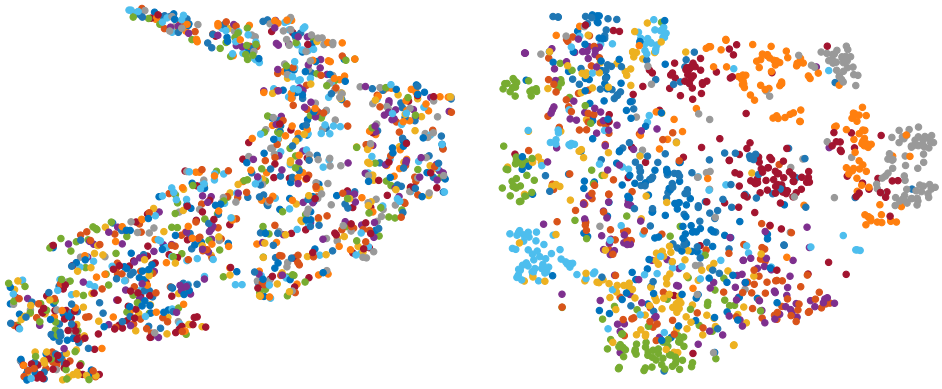
Fig 3 shows the influence of learning rate (ρ_D, ρ_G) and λ (in $L^{\text{rep}}_{D,\lambda}$, see Eq. 4) on the performance of MMD-rep and MMD-rep-b[3] on the CIFAR-10 dataset. Note that when $\lambda = -1$, MMD-rep is essentially the MMD-rbf with a single Gaussian kernel and MMD-rep-b is the same as MMD-rbf-b. We observed that: 1) both MMD-rep and MMD-rep-b performed good for $\lambda \geq 0$, with $\lambda = 0.5, 1$ slightly better than the rest; 2) the MMD-rbf model can be significantly improved by simply increasing λ from -1 to -0.5, which reduces the attraction of discriminator on real sample scores; 3) large λ may lead to unstable models when learning rate for generator ρ_G is large; 4) the RBF-B kernel managed to stabilize MMD-rep for these unstable cases but may occasionally cause the FID score to rise up. At last, it is interesting that when $\lambda > 1$, the discriminator explicitly attracts $\{D(\boldsymbol{x})\}$ and $\{D(\boldsymbol{y})\}$ via maximizing $\mathbb{E}_{P_{\mathbf{X}},P_G}[k_D(\boldsymbol{x},\boldsymbol{y})]$ (see Eq. 4). In these cases, $\mathbb{E}_{P_{\mathbf{X}},P_G}[k_D(\boldsymbol{x},\boldsymbol{y})]$ may work as a penalty on the repulsion on real samples scores that prevents the pairwise distances of $\{D(\boldsymbol{x})\}$ from increasing too fast.

5.3 Qualitative Analysis

The discriminator scores may be interpreted as a learned representation of the input samples. Fig 4 visualizes the discriminator scores learned by the MMD-rbf and proposed MMD-rep methods on CIFAR-10 dataset using t-SNE (van der Maaten (2014)). MMD-rbf ignored the class structure in data (see Fig 4a) while MMD-rep learned to concentrate the data from the same class and separate different classes to some extent (Fig 4b). This is because the discriminator D has to actively learn the data structure in order to repulse the real sample scores $\{D(\boldsymbol{x})\}$. Thus, we speculate that techniques reinforcing the learning of cluster structures in data may further improve the training of MMD-GAN.

On the other hand, the performance gain of proposed MMD-rep over the MMD-rbf and MMD-rq methods comes at no additional computational cost. In fact, by using a single kernel rather than a linear combination of kernels, MMD-rep and MMD-rep-b are simpler than MMD-rbf and MMD-rq. Besides, given a typically small batch size and a small number of discriminator output neurons (64 and 16 in our experiments), the cost of MMD over the non-saturating and hinge loss is marginal compared to the convolution operations. At last, we provide random generated samples for different methods in Appendix C.1.

[3]For $\lambda < 0$, the RBF-B kernel in Eq. 6 was used.



(a) MMD-rbf (b) MMD-rep

Figure 4: t-SNE visualization of real sample scores $\{D(\boldsymbol{x})\}$ learned by (a) MMD-rbf and (b) MMD-rep on the CIFAR-10 dataset. The samples were colored based on their class labels.

6 Discussion

This study is an extension to the previous work on MMD-GAN (Li et al. (2017a)) with two contributions. First, we interpreted the optimization of MMD loss as a combination of attraction and repulsion processes, and proposed a repulsive loss for the discriminator that actively learns the difference among real data. Second, we proposed a bounded Gaussian RBF (RBF-B) kernel that can stabilize the MMD-GAN training in many cases. We expect tuning the hyper-parameters in RBF-B or using other regularization methods may further improve our results.

The theoretical advantages of MMD-GAN require that the discriminator is injective. The proposed repulsive loss (Eq. 4) attempts to satisfy this condition by explicitly maximizing the pair-wise distances among the real samples. Li et al. (2017a) achieved the injection property by using the discriminator as the encoder and an auxiliary network as the decoder to reconstruct the real and generated samples, which is more computationally extensive than our proposed approach.

The idea of repulsion on real sample scores is in line with existing studies. It has been widely accepted that the quality of generated samples can be significantly improved by integrating labels (Odena et al. (2017); Miyato & Koyama (2018); Zhou et al. (2018)) or even pseudo-labels generated by k-means method (Grinblat et al. (2017)) in the training of discriminator. The reason may be that the labels help concentrate the data from the same class and repulse those from different classes. Using a pre-trained classifier may also help produce vivid image samples (Huang et al. (2017)) as the learned representations of the real samples in the hidden layers of the classifier tend to be well separated/organized and may produce more meaningful gradients to the generator.

At last, we note that the proposed repulsive loss is orthogonal to the GAN studies on designing network structures and training procedures, and thus may be combined with a variety of novel techniques. For example, the ResNet architecture (He et al. (2016)) has been reported to outperform the plain DCGAN used in our experiments on image generation tasks (Miyato et al. (2018); Gulrajani et al. (2017)) and self-attention module may further improve the results (Zhang et al. (2018)). On the other hand, Karras et al. (2018) proposed to progressively grows the size of both discriminator and generator and achieved the state-of-the-art performance on unsupervised training of GANs on the CIFAR-10 dataset. Future work may explore these directions.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, volume 70 of *PMLR*, pp. 214–223, 2017.

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, volume 15 of *PMLR*, pp. 215–223, 2011.

Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning, 2016. arxiv:1603.07285.

Marc G. Genton. Classes of kernels for machine learning: A statistics perspective. *J. Mach. Learn. Res.*, 2:299–312, 2002.

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

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, 2012.

Guillermo L. Grinblat, Lucas C. Uzal, and Pablo M. Granitto. Class-splitting generative adversarial networks, 2017. arXiv:1709.07359.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *NIPS*, pp. 5767–5777, 2017.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NIPS*, pp. 6626–6637, 2017.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *NIPS*, pp. 4565–4573, 2016.

Xun Huang, Yixuan Li, Omid Poursaeed, John E. Hopcroft, and Serge J. Belongie. Stacked generative adversarial networks. *CVPR*, pp. 1866–1875, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pp. 448–456, 2015.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Wei-Sheng Lai, Jia-Bin Huang, and Ming-Hsuan Yang. Semi-supervised learning for optical flow with generative adversarial networks. In *NIPS*, pp. 354–364, 2017.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabas Poczos. MMD GAN: Towards deeper understanding of moment matching network. In *NIPS*, pp. 2203–2213, 2017a.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. In *EMNLP*, pp. 2157–2169, 2017b.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *ICML*, volume 37, pp. 1718–1727, 2015.

Shuang Liu, Olivier Bousquet, and Kamalika Chaudhuri. Approximation and convergence properties of generative adversarial learning. In *NIPS*, pp. 5545–5553, 2017.

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

Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *ICLR*, 2018.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent GAN optimization is locally stable. In *NIPS*, pp. 5585–5595, 2017.

XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. On surrogate loss functions and f-divergences. *Ann. Stat.*, 37(2):876–904, 04 2009.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *ICML*, pp. 2642–2651, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *ICLR*, 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NIPS*, pp. 2234–2242, 2016.

C. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. In *ICLR*, 2017.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pp. 2818–2826, 2016.

Dustin Tran, Rajesh Ranganath, and David M. Blei. Hierarchical implicit models and likelihood-free variational inference, 2017. arXiv:1702.08896.

Thomas Unterthiner, Bernhard Nessler, Calvin Seward, Günter Klambauer, Martin Heusel, Hubert Ramsauer, and Sepp Hochreiter. Coulomb GANs: Provably optimal Nash equilibria via potential fields. In *ICLR*, 2018.

Laurens van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.

Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems Computers*, volume 2, pp. 1398–1402, 2003.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. 2015. arXiv:1506.03365.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2018. arXiv:1805.08318.

Zhiming Zhou, Han Cai, Shu Rong, Yuxuan Song, Kan Ren, Weinan Zhang, Jun Wang, and Yong Yu. Activation maximization generative adversarial nets. In *ICLR*, 2018.

J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pp. 2242–2251, 2017.

Appendices

A Stability analysis of MMD-GAN

This section demonstrates that, under mild assumptions, MMD-GAN trained by a gradient method is locally exponentially stable at equilibrium. It is organized as follows. The main assumption and proposition are presented in Section A.1, followed by simulation study in Section A.2 and proof in Section A.3. At last, we discuss the indications of assumptions on the discriminator of GAN in Section A.4.

A.1 MAIN PROPOSITION

Let $\mathcal{S}(P)$ be the support of distribution P; let $\boldsymbol{\theta}_G \in \Theta_G$, $\boldsymbol{\theta}_D \in \Theta_D$ be the parameters of the generator G and discriminator D respectively. To prove that GANs trained using the minimax loss and gradient updates is locally stable at the equilibrium point $(\boldsymbol{\theta}_D^*, \boldsymbol{\theta}_G^*)$, Nagarajan & Kolter (2017) made the following assumption:

Assumption 1 (Nagarajan & Kolter (2017))**.** $P_{\boldsymbol{\theta}_G^*} = P_{\mathbf{X}}$ *and* $\forall \boldsymbol{x} \in \mathcal{S}(P_{\mathbf{X}}), D_{\boldsymbol{\theta}_D^*}(\boldsymbol{x}) = 0$.

For loss functions like minimax and Wasserstein, $D_{\boldsymbol{\theta}_D}(\boldsymbol{x})$ may be interpreted as how plausible a sample is real. Thus at equilibrium, it may be reasonable to assume all real and generated samples are equally plausible. However, $D_{\boldsymbol{\theta}_D^*}(\boldsymbol{x}) = 0$ also indicates that $D_{\boldsymbol{\theta}_D^*}$ may have no discrimination power (see Appendix A.4 for discussion). For MMD loss in Eq. 2, $D_{\boldsymbol{\theta}_D}(\boldsymbol{x})|_{\boldsymbol{x}\sim P}$ may be interpreted as a learned representation of the distribution P. As long as two distributions P and Q match, $M^2_{k\circ D_{\boldsymbol{\theta}_D}}(P, Q) = 0$. On the other hand, $D_{\boldsymbol{\theta}_D}(\boldsymbol{x}) = 0$ is a minima solution for D but D is trained to find local maxima. Thus in contrast to Assumption 1, we assume

Assumption 2. *For GANs using MMD loss in Eq. 2 and Eq. 3, and random initialization on parameters, at equilibrium,* $D_{\boldsymbol{\theta}_D^*}(\boldsymbol{x})$ *is injective on* $\mathcal{S}(P_{\mathbf{X}}) \bigcup \mathcal{S}(P_{\boldsymbol{\theta}_G^*})$.

Assumption 2 indicates that $D_{\boldsymbol{\theta}_D^*}(\boldsymbol{x})$ is not constant almost everywhere. We use a simulation in Section A.2 to show that $D_{\boldsymbol{\theta}_D^*}(\boldsymbol{x}) = 0$ does not hold in general for MMD loss. Based on Assumption 2, we propose the following proposition and prove it in Appendix A.3:

Proposition 1. *If there exists* $\boldsymbol{\theta}_G^* \in \Theta_G$ *such that* $P_{\boldsymbol{\theta}_G^*} = P_{\mathbf{X}}$*, then GANs with MMD loss in Eq. 2 and Eq. 3 has equilibria* $(\boldsymbol{\theta}_G^*, \boldsymbol{\theta}_D)$ *for any* $\boldsymbol{\theta}_D \in \Theta_D$*. Moreover, the model trained using gradient descent methods is locally exponentially stable at* $(\boldsymbol{\theta}_G^*, \boldsymbol{\theta}_D)$ *for any* $\boldsymbol{\theta}_D \in \Theta_D$*.*

There may exist non-realizable cases where $P_{\mathbf{X}}$ cannot be represented by any generator $G_{\boldsymbol{\theta}_G}$ with $\boldsymbol{\theta}_G \in \Theta_G$. On the other hand, it is interesting to investigate the stability of the proposed repulsive loss in Eq. 4. In Section A.2, we use a simulation study to show that both the original MMD loss and the proposed repulsive loss may be locally stable and leave the proof for future work.

A.2 SIMULATION STUDY

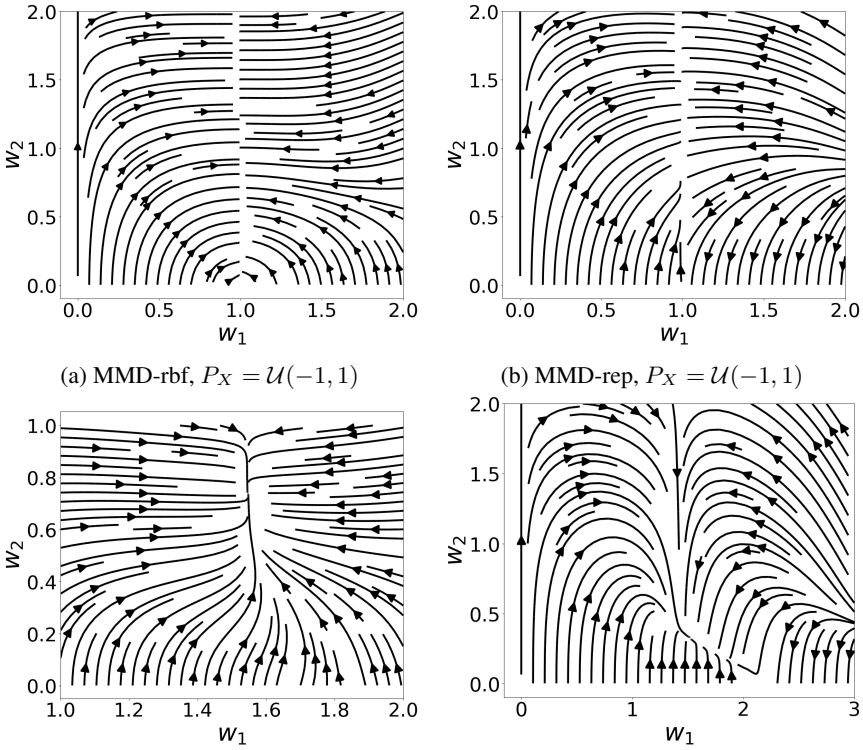
In this section, we reused the example from Nagarajan & Kolter (2017) to show that GAN using the MMD loss is locally stable. Consider a two-parameter MMD-GAN with generator $G(z) = w_1 z$, discriminator $D(x) = w_2 x^2$ and Gaussian kernel $k^{\text{rbf}}_{0.5}$. The MMD-rbf model (L_G^{mmd} and L_D^{att} from Eq. 2) and the MMD-rep model (L_G^{mmd} and $L_{D,1}^{\text{rep}}$ from Eq. 5) were tested. Each model was applied to two cases: (a) both the latent and true distributions are uniform over $[-1, 1]$ thus P_X is realizable; (b) the latent distribution is uniform over $[-1, 1]$ and the true distribution is standard Gaussian, thus P_X is non-realizable for any $w_1 \in \mathbb{R}$. Fig. S1 shows that MMD-GAN are locally stable in all cases and $D_{\boldsymbol{\theta}_D^*}(\boldsymbol{x}) = 0$ does not hold in general for MMD loss. However, MMD-rep may not be globally stable for the tested cases: initialization of (w_1, w_2) in some regions may lead to the trivial solution $w_2 = 0$ (see Fig S1b and S1d). We note that by decreasing the learning rate for G, the area of such regions decreased. At last, it is interesting to note that both MMD-rbf and MMD-rep had the same nontrivial solution $w_1 \approx 1.55$ for generator in the non-realizable cases (see Fig S1c and S1d).

A.3 PROOF OF PROPOSITION 1

This section provides the proof for Proposition 1 into two parts. First, we show that GAN with the MMD loss in Eq. 2 and Eq. 3 has equilibria for any parameter configuration of discriminator D; second, we prove the model is locally exponentially stable.

Consider real data $\mathbf{X}_r \sim P_{\mathbf{X}}$, latent variable $\mathbf{Z} \sim P_{\mathbf{Z}}$ and generated variable $\mathbf{X}_g = G_{\boldsymbol{\theta}_G}(\mathbf{Z})$. Let $\boldsymbol{x}_r$, $\boldsymbol{z}$, $\boldsymbol{x}_g$ be their instantiations. Denote $\nabla_{\boldsymbol{b}}^{\boldsymbol{a}} = \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{b}}$; $\dot{\boldsymbol{\theta}}_D = -\nabla_{\boldsymbol{\theta}_D}^{L_D}$, $\dot{\boldsymbol{\theta}}_G = -\nabla_{\boldsymbol{\theta}_G}^{L_G}$; $\boldsymbol{d}_g = D(G(\boldsymbol{z}))$, $\boldsymbol{d}_r = D(\boldsymbol{x}_r)$ where L_D and L_G are the losses for D and G respectively. Assume an isotropic stationary kernel $k(\boldsymbol{a}, \boldsymbol{b}) = k_I(\|\boldsymbol{a} - \boldsymbol{b}\|)$ (Genton (2002)) is used in MMD. We first show:

Proposition 1 (Part 1)**.** *If there exists* $\boldsymbol{\theta}_G^* \in \Theta_G$ *such that* $P_{\boldsymbol{\theta}_G^*} = P_{\mathbf{X}}$*, the GAN with the MMD loss in Eq. 2 and Eq. 3 has equilibria* $(\boldsymbol{\theta}_G^*, \boldsymbol{\theta}_D)$ *for any* $\boldsymbol{\theta}_D \in \Theta_D$*.*



(a) MMD-rbf, $P_X = \mathcal{U}(-1, 1)$

(b) MMD-rep, $P_X = \mathcal{U}(-1, 1)$

(c) MMD-rbf, $P_X = \mathcal{N}(0, 1)$

(d) MMD-rep, $P_X = \mathcal{N}(0, 1)$

Figure S1: Streamline plots of MMD-GAN using the MMD-rbf and the MMD-rep model on distributions: $P_Z = \mathcal{U}(-1, 1)$, $P_X = \mathcal{U}(-1, 1)$ or $P_X = \mathcal{N}(0, 1)$. In (a) and (b), the equilibria satisfying $P_G = P_X$ lie on the line $w_1 = 1$. In (c), the equilibrium lies around point $(1.55, 0.74)$; in (d), it is around $(1.55, 0.32)$.

Proof. Denote $\boldsymbol{e}_{i,j} = \boldsymbol{a}_i - \boldsymbol{b}_j$ and $\nabla^k_{\boldsymbol{e}_{i,j}} = \frac{\partial k(\boldsymbol{a}_i, \boldsymbol{b}_j)}{\partial \boldsymbol{e}}$ where k is the kernel of MMD. The gradients of MMD loss is

$$\dot{\boldsymbol{\theta}}_D = \mathbb{E}_{P_{\mathbf{X}}}[\nabla^k_{\boldsymbol{e}_{r1,r2}} \nabla^{\boldsymbol{e}_{r1,r2}}_{\boldsymbol{\theta}_D}] + \mathbb{E}_{P_{\boldsymbol{\theta}_G}}[\nabla^k_{\boldsymbol{e}_{g1,g2}} \nabla^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_D}] - 2\mathbb{E}_{P_{\mathbf{X}}, P_{\boldsymbol{\theta}_G}}[\nabla^k_{\boldsymbol{e}_{r,g}} \nabla^{\boldsymbol{e}_{r,g}}_{\boldsymbol{\theta}_D}] \tag{S1a}$$

$$\dot{\boldsymbol{\theta}}_G = -\mathbb{E}_{P_{\boldsymbol{\theta}_G}}[\nabla^k_{\boldsymbol{e}_{g1,g2}} (\nabla^{\boldsymbol{d}_{g1}}_{\boldsymbol{x}_{g1}} \nabla^{\boldsymbol{x}_{g1}}_{\boldsymbol{\theta}_G} - \nabla^{\boldsymbol{d}_{g2}}_{\boldsymbol{x}_{g2}} \nabla^{\boldsymbol{x}_{g2}}_{\boldsymbol{\theta}_G})] + 2\mathbb{E}_{P_{\boldsymbol{\theta}_G}, P_{\mathbf{X}}}[\nabla^k_{\boldsymbol{e}_{g,r}} \nabla^{\boldsymbol{d}_g}_{\boldsymbol{x}_g} \nabla^{\boldsymbol{x}_g}_{\boldsymbol{\theta}_G}] \tag{S1b}$$

We note that, given i.i.d. drawn samples $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^n \sim P_{\mathbf{X}}$ and $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^n \sim P_G$, an unbiased estimator of the squared MMD is (Gretton et al. (2012))

$$\hat{M}_k^2(P_{\mathbf{X}}, P_G) = \frac{1}{n(n-1)} \sum_{i \neq j}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{1}{n(n-1)} \sum_{i \neq j}^{n} k(\boldsymbol{y}_i, \boldsymbol{y}_j) - \frac{2}{n(n-1)} \sum_{i \neq j}^{n} k(\boldsymbol{x}_i, \boldsymbol{y}_j) \tag{S2}$$

At equilibrium, consider a sequence of N samples $\boldsymbol{d}_{ri} = \boldsymbol{d}_{gi} = \boldsymbol{d}_i$ with $N \to \infty$, we have

$$\dot{\boldsymbol{\theta}}_D \propto \sum_{i \neq j} \nabla^k_{\boldsymbol{e}_{i,j}} \nabla^{\boldsymbol{e}_{i,j}}_{\boldsymbol{\theta}_D} + \sum_{i \neq j} \nabla^k_{\boldsymbol{e}_{i,j}} \nabla^{\boldsymbol{e}_{i,j}}_{\boldsymbol{\theta}_D} - 2 \sum_{i \neq j} \nabla^k_{\boldsymbol{e}_{i,j}} \nabla^{\boldsymbol{e}_{i,j}}_{\boldsymbol{\theta}_D} = \mathbf{0}$$

$$\begin{aligned} \dot{\boldsymbol{\theta}}^*_G &\propto -\sum_{i \neq j} \nabla^k_{\boldsymbol{e}_{i,j}} (\nabla^{\boldsymbol{d}_i}_{\boldsymbol{x}_i} \nabla^{\boldsymbol{x}_i}_{\boldsymbol{\theta}_G} - \nabla^{\boldsymbol{d}_j}_{\boldsymbol{x}_j} \nabla^{\boldsymbol{x}_j}_{\boldsymbol{\theta}_G}) + 2 \sum_{i \neq j} \nabla^k_{\boldsymbol{e}_{i,j}} \nabla^{\boldsymbol{d}_i}_{\boldsymbol{x}_i} \nabla^{\boldsymbol{x}_i}_{\boldsymbol{\theta}_G} \\ &= \sum_{i \neq j} \nabla^k_{\boldsymbol{e}_{i,j}} (\nabla^{\boldsymbol{d}_i}_{\boldsymbol{x}_i} \nabla^{\boldsymbol{x}_i}_{\boldsymbol{\theta}_G} + \nabla^{\boldsymbol{d}_j}_{\boldsymbol{x}_j} \nabla^{\boldsymbol{x}_j}_{\boldsymbol{\theta}_G}) = \mathbf{0} \end{aligned}$$

where for $\dot{\boldsymbol{\theta}}^*_G$ we have used the fact that for each term in the summation, there exists an term with i, j reversed and $\nabla^k_{\boldsymbol{e}_{i,j}} = -\nabla^k_{\boldsymbol{e}_{j,i}}$ thus the summation is zero. Since we have not assumed the status of $\boldsymbol{\theta}_D$, $\dot{\boldsymbol{\theta}}_D = \mathbf{0}$ for any $\boldsymbol{\theta}_D \in \Theta_D$. □

We proceed to prove the model stability. First, following Theorem 5 in Gretton et al. (2012) and Theorem 4 in Li et al. (2017a), it is straightforward to see:

Lemma A.1. *Under Assumption 2, $M^2_{k\circ D_{\boldsymbol{\theta}_D}}(P_{\mathbf{X}}, P_{\boldsymbol{\theta}_G}) \geq 0$ with the equality if and only if $P_{\mathbf{X}} = P_{\boldsymbol{\theta}_G}$.*

Lemma A.1 and Proposition 1 (Part 1) state that at equilibrium $P_{\boldsymbol{\theta}^*_G} = P_{\mathbf{X}}$, every discriminator $D_{\boldsymbol{\theta}_D}$ and kernel k will give $M^2_{k\circ D_{\boldsymbol{\theta}_D}}(P_{\boldsymbol{\theta}^*_G}, P_{\mathbf{X}}) = 0$, thus no discriminator can distinguish the two distributions. On the other hand, we cite Theorem A.4 from Nagarajan & Kolter (2017):

Lemma A.2 (Nagarajan & Kolter (2017))**.** *Consider a non-linear system of parameters $(\boldsymbol{\theta}, \boldsymbol{\gamma})$: $\dot{\boldsymbol{\theta}} = h_1(\boldsymbol{\theta}, \boldsymbol{\gamma})$, $\dot{\boldsymbol{\gamma}} = h_2(\boldsymbol{\theta}, \boldsymbol{\gamma})$ with an equilibrium point at $(\mathbf{0}, \mathbf{0})$. Let there exist ϵ such that $\forall \gamma \in \mathbb{B}_\epsilon(\mathbf{0})$, $(\mathbf{0}, \boldsymbol{\gamma})$ is an equilibrium. If $\boldsymbol{J} = \frac{\partial h_1(\boldsymbol{\theta},\boldsymbol{\gamma})}{\partial \boldsymbol{\theta}}\big|_{(\mathbf{0},\mathbf{0})}$ is a Hurwitz matrix, the non-linear system is exponentially stable.*

Now we can prove:

Proposition 1 (Part 2)**.** *At equilibrium $P_{\boldsymbol{\theta}^*_G} = P_{\mathbf{X}}$, the GAN trained using MMD loss and gradient descent methods is locally exponentially stable at $(\boldsymbol{\theta}^*_G, \boldsymbol{\theta}_D)$ for any $\boldsymbol{\theta}_D \in \Theta_D$.*

Proof. Inspired by Nagarajan & Kolter (2017), we first derive the Jacobian of the system

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_{DD} & \boldsymbol{J}_{DG} \\ \boldsymbol{J}_{GD} & \boldsymbol{J}_{GG} \end{bmatrix} \triangleq \begin{bmatrix} \partial \dot{\boldsymbol{\theta}}_D^T / \partial \boldsymbol{\theta}_D & \partial \dot{\boldsymbol{\theta}}_D^T / \partial \boldsymbol{\theta}_G \\ \partial \dot{\boldsymbol{\theta}}_G^T / \partial \boldsymbol{\theta}_D & \partial \dot{\boldsymbol{\theta}}_G^T / \partial \boldsymbol{\theta}_G \end{bmatrix}$$

Denote $\Delta^{\boldsymbol{a}}_{\boldsymbol{b}} = \frac{\partial^2 \boldsymbol{a}}{\partial \boldsymbol{b}^2}$ and $\Delta^{\boldsymbol{a}}_{\boldsymbol{bc}} = \frac{\partial^2 \boldsymbol{a}}{\partial \boldsymbol{b} \partial \boldsymbol{c}}$. Based on Eq. S1, we have

$$\begin{aligned} \boldsymbol{J}_{DD} = & \mathbb{E}_{P_{\mathbf{X}}}[(\Delta^{\boldsymbol{e}_{r1,r2}}_{\boldsymbol{\theta}_D})^T \otimes (\nabla^k_{\boldsymbol{e}_{r1,r2}})^T + (\nabla^{\boldsymbol{e}_{r1,r2}}_{\boldsymbol{\theta}_D})^T \Delta^k_{\boldsymbol{e}_{r1,r2}} \nabla^{\boldsymbol{e}_{r1,r2}}_{\boldsymbol{\theta}_D}] \\ & + \mathbb{E}_{P_{\boldsymbol{\theta}_G}}[(\Delta^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_D})^T \otimes (\nabla^k_{\boldsymbol{e}_{g1,g2}})^T + (\nabla^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_D})^T \Delta^k_{\boldsymbol{e}_{g1,g2}} \nabla^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_D}] \\ & - 2\mathbb{E}_{P_{\mathbf{X}},P_{\boldsymbol{\theta}_G}}[(\Delta^{\boldsymbol{e}_{r,g}}_{\boldsymbol{\theta}_D})^T \otimes (\nabla^k_{\boldsymbol{e}_{r,g}})^T + (\nabla^{\boldsymbol{e}_{r,g}}_{\boldsymbol{\theta}_D})^T \Delta^k_{\boldsymbol{e}_{r,g}} \nabla^{\boldsymbol{e}_{r,g}}_{\boldsymbol{\theta}_D}] \end{aligned} \tag{S3a}$$

$$\begin{aligned} \boldsymbol{J}_{DG} = & \mathbb{E}_{P_{\boldsymbol{\theta}_G}}[(\Delta^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_D \boldsymbol{\theta}_G})^T \otimes (\nabla^k_{\boldsymbol{e}_{g1,g2}})^T + (\nabla^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_D})^T \Delta^k_{\boldsymbol{e}_{g1,g2}} \nabla^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_G}] \\ & - 2\mathbb{E}_{P_{\mathbf{X}},P_{\boldsymbol{\theta}_G}}[(\Delta^{\boldsymbol{e}_{r,g}}_{\boldsymbol{\theta}_D \boldsymbol{\theta}_G})^T \otimes (\nabla^k_{\boldsymbol{e}_{r,g}})^T + (\nabla^{\boldsymbol{e}_{r,g}}_{\boldsymbol{\theta}_D})^T \Delta^k_{\boldsymbol{e}_{r,g}} \nabla^{\boldsymbol{d}_g}_{\boldsymbol{\theta}_G}] \end{aligned} \tag{S3b}$$

$$\begin{aligned} \boldsymbol{J}_{GG} = & - \mathbb{E}_{P_{\boldsymbol{\theta}_G}}[(\Delta^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_G})^T \otimes (\nabla^k_{\boldsymbol{e}_{g1,g2}})^T + (\nabla^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_G})^T \Delta^k_{\boldsymbol{e}_{g1,g2}} \nabla^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_G}] \\ & + 2\mathbb{E}_{P_{\mathbf{X}},P_{\boldsymbol{\theta}_G}}[(\Delta^{\boldsymbol{d}_g}_{\boldsymbol{\theta}_G})^T \otimes (\nabla^k_{\boldsymbol{e}_{r,g}})^T + (\nabla^{\boldsymbol{d}_g}_{\boldsymbol{\theta}_G})^T \Delta^k_{\boldsymbol{e}_{r,g}} \nabla^{\boldsymbol{d}_g}_{\boldsymbol{\theta}_G}] \end{aligned} \tag{S3c}$$

where $\otimes$ is the kronecker product and $\boldsymbol{J}_{GD} = -\boldsymbol{J}_{DG}^T$. At equilibrium, consider a sequence of N samples $\boldsymbol{d}_{ri} = \boldsymbol{d}_{gi} = \boldsymbol{d}_i$ with $N \to \infty$, we have $\boldsymbol{J}_{DD} = \mathbf{0}$, $\boldsymbol{J}_{DG} = \mathbf{0}$ and

$$\begin{aligned} \boldsymbol{J}_{GG} &= \mathbb{E}_{P_{\boldsymbol{\theta}_G}}[(\nabla^{\boldsymbol{d}_{g1}}_{\boldsymbol{\theta}_G})^T \Delta^k_{\boldsymbol{e}_{g1,g2}} \nabla^{\boldsymbol{d}_{g1}}_{\boldsymbol{\theta}_G} + (\nabla^{\boldsymbol{d}_{g2}}_{\boldsymbol{\theta}_G})^T \Delta^k_{\boldsymbol{e}_{g1,g2}} \nabla^{\boldsymbol{d}_{g2}}_{\boldsymbol{\theta}_G} - (\nabla^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_G})^T \Delta^k_{\boldsymbol{e}_{g1,g2}} \nabla^{\boldsymbol{e}_{g1,g2}}_{\boldsymbol{\theta}_G}] \\ &= \mathbb{E}_{P_{\boldsymbol{\theta}_G}}[(\nabla^{\boldsymbol{d}_{g1}}_{\boldsymbol{\theta}_G})^T \Delta^k_{\boldsymbol{e}_{g1,g2}} \nabla^{\boldsymbol{d}_{g2}}_{\boldsymbol{\theta}_G} + (\nabla^{\boldsymbol{d}_{g2}}_{\boldsymbol{\theta}_G})^T \Delta^k_{\boldsymbol{e}_{g1,g2}} \nabla^{\boldsymbol{d}_{g1}}_{\boldsymbol{\theta}_G}] \end{aligned}$$

At first sight, its seems difficult to show the definite property of $\boldsymbol{J}_{GG}$. Given Lemma A.1 and fact that $\boldsymbol{J}_{GG}$ is the Hessian matrix of $M^2_{k\circ D_{\boldsymbol{\theta}_D}}(P_{\mathbf{X}}, P_{\boldsymbol{\theta}_G})$, $\boldsymbol{J}_{GG}$ is negative semidefinite. The eigenvectors of $\boldsymbol{J}_{GG}$ corresponding to zero eigenvalues form $\mathrm{null}(\boldsymbol{J}_{GG})$. There may exist small distortion $\delta\boldsymbol{\theta}_G \in \mathrm{null}(\boldsymbol{J}_{GG})$ such that $P_{\boldsymbol{\theta}^*_G + \delta\boldsymbol{\theta}_G} = P_{\boldsymbol{\theta}^*_G}$, that is, $P_{\boldsymbol{\theta}^*_G}$ is locally constant along some parameters in G. Following Lemma C.3 of Nagarajan & Kolter (2017), we consider eigenvalue decomposition $\boldsymbol{J}_{GG} = \boldsymbol{U}_G \boldsymbol{\Lambda}_G \boldsymbol{U}_G^T$ and $\boldsymbol{U}_G = [\boldsymbol{T}_G'^T, \boldsymbol{T}_G^T]$ such that $\mathrm{Col}(\boldsymbol{T}_G'^T) = \mathrm{null}(\boldsymbol{J}_{GG})$. Thus, the projections $\boldsymbol{\gamma}_G = \boldsymbol{T}_G \boldsymbol{\theta}_G$ are orthogonal to $\mathrm{null}(\boldsymbol{J}_{GG})$. Then, the Jacobian corresponding to the projected system has the form $\boldsymbol{J}' = [\mathbf{0}, \mathbf{0}; \mathbf{0}, \boldsymbol{J}'_{GG}]$ with negative definite block $\boldsymbol{J}'_{GG} = \boldsymbol{T}_G \boldsymbol{J}_{GG} \boldsymbol{T}_G^T$ and zeros elsewhere. Moreover, on all directions exclude those described by $\boldsymbol{J}'_{GG}$, the system is surrounded by a neighborhood of equilibia at least locally. According to Lemma A.2, the system is exponentially stable. □

A.4 DISCUSSION ON ASSUMPTION 1

This section shows that constant discriminator output $D_{\boldsymbol{\theta}_D^*}(\boldsymbol{x}) = \boldsymbol{c}$ for $\boldsymbol{x} \in \mathcal{S}(P_{\mathbf{X}}) \bigcup \mathcal{S}(P_{\boldsymbol{\theta}_G^*})$ indicates that $D_{\boldsymbol{\theta}_D^*}$ may have no discrimination power. First, we make the following assumptions:

Assumption 3. *1. D is a multilayer perceptron where each layer l can be factorized into an affine transform and an element-wise activation function f_l. 2. Each activation function $f_l \in C^0$; furthermore, f_l' has a finite number of discontinuities and $f_l'' \in C^0$[4]. 3. Input data to D is continuous and its support $\mathcal{S}$ is compact in $\mathbb{R}^d$ with non-zero measure in each dimension and $d > 1$[5].*

Based on Assumption 3, we have the following proposition:

Proposition 2. *If $\forall \boldsymbol{x} \in \mathcal{S}$, $D(\boldsymbol{x}) = \boldsymbol{c}$, where $\boldsymbol{c}$ is constant, then there always exists distortion $\delta\boldsymbol{x}$ such that $\boldsymbol{x} + \delta\boldsymbol{x} \notin \mathcal{S}$ and $D(\boldsymbol{x} + \delta\boldsymbol{x}) = \boldsymbol{c}$.*

Proof. Without loss of generality, we consider $D(\boldsymbol{x}) = \boldsymbol{W}_2 h(\boldsymbol{x}) + \boldsymbol{b}_2$ and $h(\boldsymbol{x}) = f(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1)$, where $\boldsymbol{W}_1 \in \mathbb{R}^{d_h \times d}$, $\boldsymbol{W}_2$, $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ are model weights and biases, f is some activation function. For $\boldsymbol{x} \in \mathcal{S}$, since $D(\boldsymbol{x}) = c$, we have $h(\boldsymbol{x}) \in \mathrm{null}(\boldsymbol{W}_2)$. Furthermore:

(a) If $\mathrm{rank}(\boldsymbol{W}_1) < d$, for any $\delta\boldsymbol{x} \in \mathrm{null}(\boldsymbol{W}_1)$, $h(\boldsymbol{x} + \delta\boldsymbol{x}) \in \mathrm{null}(\boldsymbol{W}_2)$.

(b) If $\mathrm{rank}(\boldsymbol{W}_1) = d = d_h$, the problem $h(\boldsymbol{x} + \delta\boldsymbol{x}) = k \cdot h(\boldsymbol{x})$ has unique solution for any $k \in \mathbb{R}$ as long as $k \cdot h(\boldsymbol{x})$ is within the output range of f.

(c) If $\mathrm{rank}(\boldsymbol{W}_1) = d < d_h$, let $\boldsymbol{U}$ and $\boldsymbol{V}$ be two basis matrices of R^{d_h} such that $\boldsymbol{W}_1\boldsymbol{x} = \boldsymbol{U}\begin{bmatrix}\hat{\boldsymbol{x}}^T & \boldsymbol{0}^T\end{bmatrix}^T$ and any vector in $\mathrm{null}(\boldsymbol{W}_2)$ can be represented as $\boldsymbol{V}\begin{bmatrix}\boldsymbol{z}^T & \boldsymbol{0}^T\end{bmatrix}^T$, where $\hat{\boldsymbol{x}} \in \mathbb{R}^{d_h \times d}$, $\boldsymbol{z} \in \mathbb{R}^{d_h \times n}$ and n is the nullity of $\boldsymbol{W}_2$. Let the projected support be $\hat{\mathcal{S}}$. Thus, $\forall \hat{\boldsymbol{x}} \in \hat{\mathcal{S}}$, there exists $\boldsymbol{z}$ such that $f(\boldsymbol{U}\begin{bmatrix}\hat{\boldsymbol{x}}^T & \boldsymbol{0}^T\end{bmatrix}^T + \boldsymbol{b}_1) = \boldsymbol{V}\begin{bmatrix}\boldsymbol{z}^T & \boldsymbol{z}_c^T\end{bmatrix}^T$ with $\boldsymbol{z}_c = \boldsymbol{0}$. Consider the Jacobian:

$$\boldsymbol{J} = \frac{\partial \begin{bmatrix}\boldsymbol{z}^T & \boldsymbol{z}_c^T\end{bmatrix}^T}{\partial \begin{bmatrix}\hat{\boldsymbol{x}}^T & \boldsymbol{0}^T\end{bmatrix}^T} = \boldsymbol{V}^{-1}\nabla\boldsymbol{\Sigma}\boldsymbol{U} \tag{S4}$$

where $\nabla\boldsymbol{\Sigma} = \mathrm{diag}(\frac{\mathrm{d}\,f}{\mathrm{d}\,a_i})$ and $\boldsymbol{a} = [a_i]_{i=1}^{d_h}$ is the input to activation, or pre-activations. Since $\hat{\mathcal{S}}$ is continuous and compact, it has infinite number of boundary points $\{\hat{\boldsymbol{x}}_b\}$ for $d > 1$. Consider one boundary point $\hat{\boldsymbol{x}}_b$ and its normal line $\delta\hat{\boldsymbol{x}}_b$. Let $\epsilon > 0$ be a small scalar such that $\hat{\boldsymbol{x}}_b - \epsilon\delta\hat{\boldsymbol{x}}_b \in \hat{\mathcal{S}}$ and $\hat{\boldsymbol{x}}_b + \epsilon\delta\hat{\boldsymbol{x}}_b \notin \hat{\mathcal{S}}$.

- For linear activation, $\nabla\boldsymbol{\Sigma} = \boldsymbol{I}$ and $\boldsymbol{J}$ is constant. Then $\boldsymbol{z}_c$ remains $\boldsymbol{0}$ for $\hat{\boldsymbol{x}}_b + \epsilon\delta\hat{\boldsymbol{x}}_b$, i.e., there exists $\boldsymbol{z}$ such that $h(\hat{\boldsymbol{x}} + \epsilon\delta\hat{\boldsymbol{x}}) \in \mathrm{null}(\boldsymbol{W}_2)$.
- For nonlinear activations, assume f' has N discontinuities. Since $\boldsymbol{U}\begin{bmatrix}\hat{\boldsymbol{x}}^T & \boldsymbol{0}^T\end{bmatrix}^T + \boldsymbol{b}_1 = \boldsymbol{c}$ has unique solution for any vector $\boldsymbol{c}$, the boundary points $\{\hat{\boldsymbol{x}}_b\}$ cannot yield pre-activations $\{\boldsymbol{a}_b\}$ that all lie on the discontinuities in any of the d_h directions. Though we might need to sample d_h^{N+1} points in the worst case to find an exception, there are infinite number of exceptions. Let $\hat{\boldsymbol{x}}_b$ be a sample where $\{\boldsymbol{a}_b\}$ does not lie on the discontinuities in any direction. Because f'' is continuous, $\boldsymbol{z}_c$ remains $\boldsymbol{0}$ for $\hat{\boldsymbol{x}}_b + \epsilon\delta\hat{\boldsymbol{x}}_b$, i.e., there exists $\boldsymbol{z}$ such that $h(\hat{\boldsymbol{x}} + \epsilon\delta\hat{\boldsymbol{x}}) \in \mathrm{null}(\boldsymbol{W}_2)$.

In conclusion, we can always find $\delta\boldsymbol{x}$ such that $\boldsymbol{x} + \delta\boldsymbol{x} \notin \mathcal{S}$ and $D(\boldsymbol{x} + \delta\boldsymbol{x}) = \boldsymbol{c}$. □

Proposition 2 indicates that if $D_{\boldsymbol{\theta}_D^*}(\boldsymbol{x}) = \boldsymbol{0}$ for $\boldsymbol{x} \in \mathcal{S}(P_{\mathbf{X}}) \bigcup \mathcal{S}(P_{\boldsymbol{\theta}_G^*})$, $D_{\boldsymbol{\theta}_D^*}$ cannot discriminate against fake samples with distortions to the original data. In contrast, Assumption 2 and Lemma A.1 guarantee that, at equilibrium, the discriminator trained using MMD loss function is effective against such fake samples given a large number of i.i.d. test samples (Gretton et al. (2012)).

[4] This include many commonly used activations like linear, sigmoid, tanh, ReLU and ELU.

[5] For distributions with semi-infinite or infinite support, we consider the effective or truncated support $\mathcal{S}_\epsilon(P) = \{x \in \mathcal{X} | P(x) \geq \epsilon\}$, where $\epsilon > 0$ is a small scalar. This is practical, e.g., univariate Gaussian has support in $(-\infty, +\infty)$ yet a sample five standard deviations away from the mean is unlikely to be valid.

B SUPPLEMENTARY ON METHODOLOGY

B.1 POWER ITERATION FOR CONVOLUTION OPERATION

This section proposes the power iteration for convolution operation (PICO) method to estimate the spectral norm of a convolution kernel.

For a weight matrix $\boldsymbol{W}$, the spectral norm is defined as $\sigma(\boldsymbol{W}) = \max_{\|\boldsymbol{v}\|_2 \leq 1} \|\boldsymbol{W}\boldsymbol{v}\|_2$. Power iteration on matrix (PIM) is often used to estimate $\sigma(\boldsymbol{W})$, which iterates between the following two steps:

1. Update $\boldsymbol{u} = \boldsymbol{W}\boldsymbol{v} / \|\boldsymbol{W}\boldsymbol{v}\|_2$;
2. Update $\boldsymbol{v} = \boldsymbol{W}^T\boldsymbol{u} / \left\|\boldsymbol{W}^T\boldsymbol{u}\right\|_2$.

The convolutional kernel $\boldsymbol{W}_c$ is a tensor of shape $h \times w \times c_{in} \times c_{out}$ with h, w the receptive field size and c_{in}, c_{out} the number of input/output channels. To estimate $\sigma(\boldsymbol{W}_c)$, Miyato et al. (2018) reshaped it into a matrix $\boldsymbol{W}_{rs}$ of shape $(hwc_{in}) \times c_{out}$ and estimated $\sigma(\boldsymbol{W}_{rs})$.

We propose a simple method to calculate $\boldsymbol{W}_c$ directly based on the fact that convolution operation is linear. For any linear map $T : \mathbb{R}^m \to \mathbb{R}^n$, there exists matrix $\boldsymbol{W}_L \in \mathbb{R}^{n \times m}$ such that $\boldsymbol{y} = T(\boldsymbol{x})$ can be represented as $\boldsymbol{y} = \boldsymbol{W}_L\boldsymbol{x}$. Thus, we may simply substitute $\boldsymbol{W}_L = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$ in the PIM method to estimate the spectral norm of any linear operation. In the case of convolution operation $*$, there exist doubly block circulant matrix $\boldsymbol{W}_{dbc}$ such that $\boldsymbol{u} = \boldsymbol{W}_c * \boldsymbol{v} = \boldsymbol{W}_{dbc}\boldsymbol{v}$. Consider $\boldsymbol{v}' = \boldsymbol{W}_{dbc}^T\boldsymbol{u} = [\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{v}}]^T\boldsymbol{u}$ which is essentially the transpose convolution of $\boldsymbol{W}_c$ on $\boldsymbol{u}$ (Dumoulin & Visin (2016)). Thus, similar to PIM, PICO iterates between the following two steps:

1. Update $\boldsymbol{u} = \boldsymbol{W}_c * \boldsymbol{v} / \|\boldsymbol{W}_c * \boldsymbol{v}\|_2$;
2. Do transpose convolution of $\boldsymbol{W}_c$ on $\boldsymbol{u}$ to get $\hat{\boldsymbol{v}}$; update $\boldsymbol{v} = \hat{\boldsymbol{v}} / \|\hat{\boldsymbol{v}}\|_2$.

Empirically we found that for convolution kernel $\boldsymbol{W}_c$, $\sigma_{\text{PICO}}(\boldsymbol{W}_c) > \sigma_{\text{PIM}}(\boldsymbol{W}_c)$. That is, spectral normalization using PICO imposes a stronger constraint than PIM. However, we did not find significant and consistent difference in performance between PICO and PIM on tested datasets. Nevertheless, we proposed a general framework to estimate the spectral norm for any linear operation (and possibly combination of linear operations), which may be helpful in more general applications.

B.2 NETWORK ARCHITECTURE

For unsupervised image generation tasks on CIFAR-10 and STL-10 datasets, the DCGAN (Radford et al. (2016)) architecture from Miyato et al. (2018) was used. For CelebA and LSUN bedroom datasets, we added more layers to the generator and discriminator accordingly. See Table S1 and S2 for details.

C SUPPLEMENTARY ON EXPERIMENTS

C.1 UNSUPERVISED IMAGE GENERATION

Generated samples on CelebA dataset are given in Fig. S2 and LSUN bedrooms in Fig. S3.

Table S1: DCGAN models for image generation on CIFAR-10 ($h = w = 4$, $H = W = 32$) and STL-10 ($h = w = 6$, $H = W = 48$) datasets. For non-saturating loss and hinge loss, $s = 1$; for MMD-rand, MMD-rbf, MMD-rq, $s = 16$.

(a) Generator

$\boldsymbol{z} \in \mathbb{R}^{128} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$128 \to h \times w \times 512$, dense, linear
4×4, stride 2 deconv, 256, BN, ReLU
4×4, stride 2 deconv, 128, BN, ReLU
4×4, stride 2 deconv, 64, BN, ReLU
3×3, stride 1 conv, 3, Tanh

(b) Discriminator

RGB image $\boldsymbol{x} \in [-1, 1]^{H \times W \times 3}$
3×3, stride 1 conv, 64, LReLU
4×4, stride 2 conv, 128, LReLU 3×3, stride 1 conv, 128, LReLU
4×4, stride 2 conv, 256, LReLU 3×3, stride 1 conv, 256, LReLU
4×4, stride 2 conv, 512, LReLU 3×3, stride 1 conv, 512, LReLU
$h \times w \times 512 \to s$, dense, linear

Table S2: DCGAN models for image generation on CelebA and LSUN-bedroom datasets. For non-saturating loss and hinge loss, $s = 1$; for MMD-rand, MMD-rbf, MMD-rq, $s = 16$.

(a) Generator

$\boldsymbol{z} \in \mathbb{R}^{128} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$128 \to 4 \times 4 \times 1024$, dense, linear
4×4, stride 2 deconv, 512, BN, ReLU
4×4, stride 2 deconv, 256, BN, ReLU
4×4, stride 2 deconv, 128, BN, ReLU
4×4, stride 2 deconv, 64, BN, ReLU
3×3, stride 1 conv, 3, Tanh

(b) Discriminator

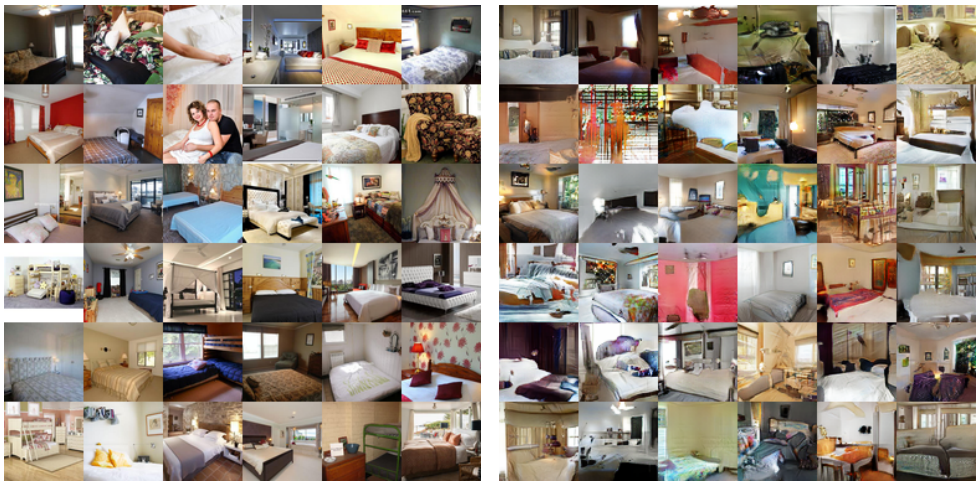
RGB image $\boldsymbol{x} \in [-1, 1]^{64 \times 64 \times 3}$
3×3, stride 1 conv, 64, LReLU
4×4, stride 2 conv, 128, LReLU 3×3, stride 1 conv, 128, LReLU
4×4, stride 2 conv, 256, LReLU 3×3, stride 1 conv, 256, LReLU
4×4, stride 2 conv, 512, LReLU 3×3, stride 1 conv, 512, LReLU
4×4, stride 2 conv, 1024, LReLU 3×3, stride 1 conv, 1024, LReLU
$4 \times 4 \times 512 \to s$, dense, linear

(a) Real samples (b) Hinge

(c) MMD-rbf (d) MMD-rbf-b

(e) MMD-rep (f) MMD-rep-b

Figure S2: Image generation using different loss functions on 64×64 CelebA dataset.

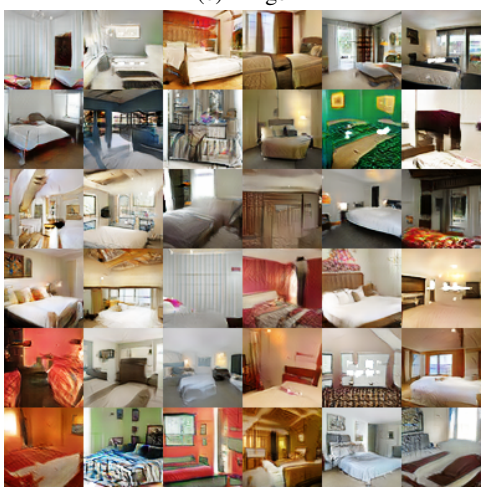


(a) Real samples

(b) Non-saturating

(c) Hinge

(d) MMD-rbf-b



(e) MMD-rep

(f) MMD-rep-b

Figure S3: Image generation using different loss functions on 64×64 LSUN bedroom dataset.