Geometric GAN

Jae Hyun Lim[1], **Jong Chul Ye**[2,3]
[1] ETRI, South Korea
jaehyun.lim@etri.re.kr
[2] Dept. of Bio and Brain engineering, KAIST, South Korea
[3] Dept. of Mathematical Sciences, KAIST, South Korea
jong.ye@kaist.ac.kr

Abstract

Generative Adversarial Nets (GANs) represent an important milestone for effective generative models, which has inspired numerous variants seemingly different from each other. One of the main contributions of this paper is to reveal a unified geometric structure in GAN and its variants. Specifically, we show that the adversarial generative model training can be decomposed into three geometric steps: separating hyperplane search, discriminator parameter update away from the separating hyperplane, and the generator update along the normal vector direction of the separating hyperplane. This geometric intuition reveals the limitations of the existing approaches and leads us to propose a new formulation called *geometric GAN* using SVM separating hyperplane that maximizes the margin. Our theoretical analysis shows that the geometric GAN converges to a Nash equilibrium between the discriminator and generator. In addition, extensive numerical results show that the superior performance of geometric GAN.

1 Introduction

Recently, inspired by the success of the deep discriminative models, Goodfellow et al [1] proposed a novel generative model training method called generative adversarial nets (GAN). GAN is formulated as a minimax game between a generative network (generator) that maps a random vector into the data space and a discriminative network (discriminator) trying to distinguish the generated samples from real samples. Unlike the classical generative models such as Variational Auto-Encoders (VAEs) [2], the minimax formation of GAN can transfer the success of deep discriminative models to generative models, resulting in significant improvement in generative model performance [1].

Specifically, the original form of the GAN solves the following minmax game:

$$\min_G \max_D L_{GAN}(D,G) \tag{1}$$

where

$$L_{GAN}(D,G) \quad := \quad E_{x\sim P_S}\left[\log D(x)\right] + E_{z\sim P_Z}\left[\log(1-D(G(z))\right], \tag{2}$$

where P_S is the sample distribution; $D(x)$ is the discriminator that takes $x \in \mathcal{X}$ as input and outputs a scalar between $[0,1]$; $G(z)$ is the generator that maps a sample z drawn from a distribution P_Z to the input space $\mathcal{X}$. The meaning of (1) is that the generator tries to fool out the discriminator while the discriminator wants to maximize the differentiation power between the true and generated samples. The authors further showed that the GAN training is indeed to approximate the minimization of the symmetric Jensen-Shannon divergence [1]. This idea has been generalized by the authors in [3] for all f-divergences. Moreover, Maximum Mean Discrepancy objective (MMD) for GAN training was also proposed in [4, 5].

It is well-known that the training GAN is difficult. In particular, the authors in [6] have identified the following sources of the difficulties: 1) when the discriminator becomes accurate, the

当判别器准确性很高时，很难将判别器的成功转移到生成器中

gradient for generator vanishes, 2) a popular fixation using a generator gradient updating with $E_{z\sim P_Z}[-\log D(G(z))]$ is unstable because of the singularity at the denominator when the discriminator is accurate. The main motivation of Wasserstein GAN (W-GAN) [7] was, therefore, to introduce the weight clipping to address the above-described limitations. In fact, Wasserstein GAN is a special instance of minimizing the integral probability metric (IPM) [8], and Mroueh et al [9] recently generalized the W-GAN for wider function classes and proposed the mean feature matching and/or covariance feature matching GAN (McGAN) using the IPM minimization framework [9].

Inspired by McGAN, here we propose a novel geometric generalization called *geometric GAN*. Specifically, geometric GAN is inspired by our novel observation that McGAN is composed of three geometric operations in feature space:

- **Separating hyperplane search**: finding the separating hyperplane for a linear classifier [10, 11, 12]
- **Discriminator update away from the hyperplane:** discriminator parameter update *away from* the separating hyperplane using stochastic gradient direction (SGD).
- **Generator update toward the hyperplane:** generator parameter update along the normal vector direction of the separating hyperplane using stochastic gradient direction (SGD).

IPM 衡量在特征空间中均值向量的最大投影距离

This geometric interpretation proves to be very general, so it can be applied to most of the existing GAN and its variants. Indeed, the main differences between the algorithms come from the construction of the separating hyperplanes for a linear classifier on feature space and the geometric scaling factors for the feature vectors. Based on this observation, we provide new geometric interpretations of GAN [1], f-GAN [3], EB-GAN [13], and W-GAN [7] in terms of separating hyperplanes and geometric scaling factors, and discuss their limitations. Furthermore, we propose a novel *geometric GAN* using the support vector machine (SVM) separating hyperplane that has maximal margin between two classes of separable data [14, 11]. Our numerical experiments clearly showed that the proposed geometric GAN outperforms the existing GANs in all data set.

每次寻找超平面，其所在的特征空间都不一样了

2 Related Approaches

In order to introduce the geometric interpretation of GAN and its variants, we begin with the review of the mean feature matching GAN (McGAN) [9].

2.1 Mean feature matching GAN

Let $\mathcal{F}$ be a set of bounded real valued functions on the sample space $\mathcal{X}$. Suppose that P and Q are two probability distributions on $\mathcal{X}$. Then, the integral probability metric (IPM) between P and Q on the function space $\mathcal{F}$ is defined as follows [8, 15, 16]:

$$\begin{aligned} d_{\mathcal{F}}(P,Q) &= \sup_{f\in\mathcal{F}} \left| \int f dP - \int f dQ \right| \\ &= \sup_{f\in\mathcal{F}} |E_{x\sim P}[f(x)] - E_{x\sim Q}[f(x)]| \end{aligned} \tag{3}$$

where $E_{x\sim P}[\cdot]$ denotes the expectation with respect to the probability distribution $P(x)$. We can easily show that $d_{\mathcal{F}}$ is non-negative, symmetric and satisfies the triangle inequality. So $d_{\mathcal{F}}$ can be used as a distance measure in the probability space. For example, when $\mathcal{F}$ is defined as a collection of functions with a finite Lipschitz constant, the IPM is the Wasserstein distance or the earth mover's distance [15, 16] that forms the basis of the Wasserstein GAN [7].

In McGAN, the generator network $g_\theta : \mathcal{Z} \to \mathcal{X}$ that maps a random input $z \in \mathcal{Z}$ to a target $x \in \mathcal{X}$ is shown as a block in Fig. 1(a), and the function space $\mathcal{F}$ under study is defined as follows [9]:

$$\mathcal{F}_{w,\zeta} = \{f(x) = \langle w, \Phi_\zeta(x)\rangle \mid w \in \Xi, \|w\|_2 \le 1\}$$

where $\Phi_\zeta : \mathcal{X} \to \Xi$ is a bounded map from $\mathcal{X}$ to a (often higher-dimensional) *feature space* Ξ. Note that $\mathcal{F}_{w,\zeta}$ is a symmetric function spaces because if $f \in \mathcal{F}_{w,\zeta}$, then $-f \in \mathcal{F}_{w,\zeta}$. Then, the IPM between P_x and P_g is given by

$$d_{\mathcal{F}}(P_x, P_g) = \max_{\|w\|_2\le 1} \langle w, E_{x\sim P_x}\Phi_\zeta(x) - E_{g_\theta\sim P_g}\Phi_\zeta(g_\theta)\rangle$$

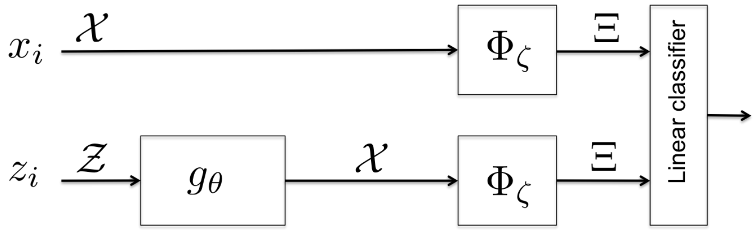


Figure 1: Structure of the mean feature matching GAN and its extension to geometric GAN.

under the constraints that Φ_ζ is bounded and $\|w\|_2 \leq 1$.

投影向量的模小于等于1 指定的约束，那最终刻画的是均值向量差的模

Now, given a finite sequence of mini-batch training data set $S = \{(z_1, x_1), \cdots, (z_n, x_n)\}$, an empirical estimate of the minmax game that minimizes the $d_{\mathcal{F}}(P_x, P_g)$ is given by [9]:

$$\min_{\theta} \max_{\|w\|_2 \leq 1, \zeta} \quad \hat{L}(w, \zeta, \theta) \tag{4}$$

$$\tag{5}$$

where

$$\hat{L}(w, \zeta, \theta) := \left\langle w, \frac{1}{n}\sum_{i=1}^{n} \Phi_\zeta(x_i) - \frac{1}{n}\sum_{i=1}^{n} \Phi_\zeta(g_\theta(z_i)) \right\rangle$$

Then, the discriminator update can be done using a stochastic gradient descent (SGD) [9]:

$$(w, \zeta) \quad \leftarrow \quad (w, \zeta) + \eta \left(\nabla_w \hat{L}(w, \zeta, \theta), \nabla_\zeta \hat{L}(w, \zeta, \theta) \right) \tag{6}$$

where η is a learning rate. The authors in [9] also used the projection onto the unit l_p ball and weight clipping for w and ζ updates, respectively, to meet the constraints. Using the updated w, the generator update is then given by [9]:

$$\theta \quad \leftarrow \quad \theta + \eta \sum_{i=1}^{n} \langle w, \nabla_\theta \Phi(g_\theta(z_i)) \rangle / n \,. \tag{7}$$

2.2 Geometric interpretation of the mean feature matching GAN

The primal form of the w update in (6) is geometrically less informative, so here we use the Cauchy-Swartz inequality to obtain a closed-form update for w:

$$w^* = c \sum_{i=1}^{n} \left(\Phi_\zeta(x_i) - \Phi_\zeta(g_\theta(z_i)) \right) / n \tag{8}$$

where the constant c is given by $c = \| \sum_{i=1}^{n} \left(\Phi_\zeta(x_i) - \Phi_\zeta(g_\theta(z_i)) \right) / n \|^{\frac{1}{2}}$. Given w^*, the corresponding discriminator and generator updates are represented by

$$\zeta \quad \leftarrow \quad \zeta + \eta \sum_{i=1}^{n} \langle w^*, \nabla_\zeta \Phi_\zeta(x_i) \;\; - \nabla_\zeta \Phi_\zeta(g_\theta(z_i)) \rangle / n \tag{9}$$

$$\theta \quad \leftarrow \quad \theta + \eta \sum_{i=1}^{n} \langle w^*, \nabla_\theta \Phi_\zeta(g_\theta(z_i)) \rangle / n \tag{10}$$

Note that the update equations (8) to (10) are equivalent to the dual form of the McGAN [9], where the authors derived the following minmax problem by using (8):

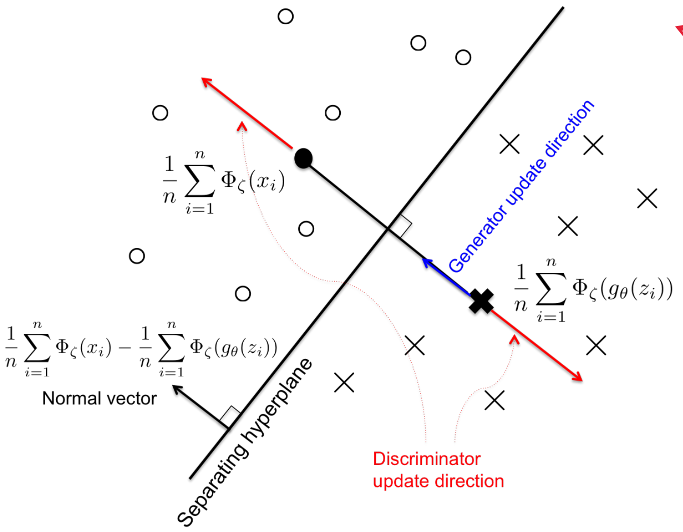
IPM框架下距离的对偶形式，注意w是模小于等于1的方向向量，内积最大，最后的结果就是均值向量的欧式距离

$$\min_\theta \max_\zeta \frac{1}{2} \left\| \sum_{i=1}^{n} \Phi_\zeta(x_i) - \sum_{i=1}^{n} \Phi_\zeta(g_\theta(z_i)) \right\|^2 \,.$$

However, we notice that the explicit representation by Eqs. (8) to (10) gives clearer geometric intuition that plays the key role in designing a geometric GAN, ass will become clear soon.

判别器 maximal discriminate

More specifically, in designing a linear classifier for two class classification problems, (8) is known as the normal vector for the separating hyperplane for the *mean difference* (MD) classifier [10, 11, 12]. As shown in Fig. 2, once the separating hyperplane is defined, the SGD udpate (9) is to update the discriminator parameters such that the true and fake samples are maximally separately away from the separating hyperplane parallel to the normal vector. On the other hand, the SGD udpate using (10) is to update the generator parameters to make the fake samples approach the separating hyperplane along the normal vector direction (see Fig. 2).


$\frac{1}{n}\sum_{i=1}^{n}\Phi_\varsigma(x_i)$
$\frac{1}{n}\sum_{i=1}^{n}\Phi_\varsigma(g_\theta(z_i))$
Generator update direction
$\frac{1}{n}\sum_{i=1}^{n}\Phi_\varsigma(x_i) - \frac{1}{n}\sum_{i=1}^{n}\Phi_\varsigma(g_\theta(z_i))$
Normal vector
Separating hyperplane
Discriminator update direction

问题是这种距离在训练过程中是否是稳定下降呢

Figure 2: Geometry of the mean feature matching GAN.

Recall that a linear classifier is defined via the normal vector to the separating hyperplane and an offset [14]. Thus, comparing the direction between two classifiers means comparing their normal vector directions. As shown in Appendix, aside from the geometric scaling factors, the existing GAN and its variants mainly differ in their definition of the normal vector for the separating hyperplane. Based on this observation, in the next section, we discuss an optimal separating hyperplane for generative model training that has the maximal margin.

3 Geometric GAN

3.1 Linear classifiers in high-dimension low-sample size

In adversarial training in feature space, a discriminator is interested in discriminating true samples $\{\Phi_\varsigma(x_i)\}_{i=1}^n$ and the fake samples $\{\Phi_\varsigma(g_\theta(z_i))\}_{i=1}^n$. In practice, the minibatch size n is much smaller than the dimension of the feature space d, and this type of classification problem is often called the high-dimension low-sample size (HDLSS) problem [10, 11, 12].

In fact, the mean difference (MD) classifier is one of the popular methods for HDLSS. Specifically, the MD classifier selects the hyperplane that lies half way between the two class means. In particular the normal vector w for the seperating hyperplane is given by the difference of the class means:

$$w^{MD} = \frac{1}{n}\sum_{i=1}^{n}\Phi_\varsigma(x_i) - \frac{1}{n}\sum_{i=1}^{n}\Phi_\varsigma(g_\theta(z_i)) \qquad (12)$$

Note that if the variables are first mean centered then scaled by the standard deviation, then the mean difference is equivalent to the naive Bayes classifier [10, 11, 12]. Furthermore, in HDLSS, there always exists a maximal data piling direction (MDP) [12], where mulitple points in each class have the identical projection on the line spanned by the normal vector.

On the other hand, the Support Vector Machine (SVM) [14] and its many variants is one of the most widely used and well studied classification algorithms and its robustness has been also proven for HDLSS setup [10, 11, 12]. Although the aforementioned classification algorithms are motivated by fitting a *statistical* distribution to the data, SVM is motivated by a *geometric* heuristic that leads directly to an optimization problem: maximize the margin between two classes of separable data. In addition, soft-margin SVM balances two competing objectives to maximize the margin while penalizing points on the wrong side of the margin.

Recently, Carmichael et al [11] investigated the Karush-Kuhn-Tucker conditions to provide rigorous mathematical proof for new insights into the behaviour of soft-margin SVM in the large and small tuning parameter regimes in HDLSS. They revealed that for small tuning parameter, if the number of data in two classes are the same (which is the case in our problem), then the SVM direction becomes exactly the MD direction. In addition, for sufficiently large tuning parameter, the authors showed that soft margin SVM is equivalent to hard margin SVM if the data are separable, and the hard-margin SVM has data piling. Due to this generality of the soft-margin SVM, the proposed geometric GAN is designed based on soft-margin SVM linear classifier.

3.2 Geometric GAN with SVM hyperplane

Note that soft-margin SVM is designed by adding a tunning parameter C and slack variables ξ_i which allows points to be on the wrong side of the margin [14]. In our problem classifying the true samples versus fake samples, the primal form of soft-margin SVM can be formulated by

$$\begin{aligned} \min_{w,b} \quad & \tfrac{1}{2}\|w\|^2 + C\textstyle\sum_i(\xi_i+\xi_i') \\ \text{subject to} \quad & \langle w, \Phi_\zeta(x_i)\rangle + b \geq 1-\xi_i, \qquad i=1,\cdots,n \\ & \langle w, \Phi_\zeta(g_\theta(z_i))\rangle + b \leq \xi_i' - 1, \quad i=1,\cdots,n \\ & \xi_i, \xi_i' \geq 0, \quad i=1,\cdots,n \end{aligned}$$

Equivalently, the primal form of the soft-margin SVM can be represented using loss + penalty form [17, 14]:

$$\min_{w,b} R_\theta(w,b;\zeta) \tag{13}$$

where

$$\begin{aligned} R_\theta(w,b;\zeta) \quad = \quad & \frac{1}{2Cn}\|w\|^2 + \frac{1}{n}\sum_{i=1}^n \max\left(0, 1-\langle w, \Phi_\zeta(x_i)\rangle - b\right) \\ & + \frac{1}{n}\sum_{i=1}^n \max\left(0, 1+\langle w, \Phi_\zeta(g_\theta(z_i))\rangle + b\right) \end{aligned} \tag{14}$$

The goal of the SVM optimization (13) is to maximize the margin between the two classes. This implies that the discriminator update can be also easily incorporated with SVM update, because the goal of the discriminator update can be also regarded to maximize the margin between the two classes. More specifically, our optimization problem is given by

$$\min_{w,b,\zeta} R_\theta(w,b;\zeta) \tag{15}$$

for a given generator parameter θ.

Specifically, in SVM, the normal vector for the optimal separating hyperplane w^{SVM} from (13) is given by [11, 14]:

$$w^{SVM} := \sum_{i=1}^n \alpha_i \Phi_\zeta(x_i) - \sum_{i=1}^n \beta_i \Phi_\zeta(g_\theta(z_i)) \tag{16}$$

where (α_i, β_i) will be nonzero only for the support vectors, where the set of support vectors now includes all data points on the margin boundary as well as those on the wrong side of the margin boundary (see Fig. 3). More specifically, we define the region $\mathcal{M}$ between the margin boudaries as shown in Fig. 3(a):

$$\mathcal{M} \quad = \quad \left\{\phi \in \Xi \mid |\langle w^{SVM}, \phi\rangle + b| \leq 1\right\}. \tag{17}$$

Then, for given w^{SVM} and b, the cost function (14) then becomes

$$\begin{aligned} R_\theta(w,b;\zeta) &:= \frac{1}{n}\sum_{i\in I_S}\langle w^{SVM}, \Phi_\zeta(g_\theta(z_i))\rangle - \frac{1}{n}\sum_{i\in I_T}\langle w^{SVM}, \Phi_\zeta(x_i)\rangle + \text{constant} \\ &= \frac{1}{n}\sum_{i=1}^{n}\langle w^{SVM}, s_i\Phi_\zeta(g_\theta(z_i)) - t_i\Phi_\zeta(x_i)\rangle + \text{constant} \end{aligned} \tag{18}$$

where (t_i, s_i) are *geometric scaling factors* defined by

$$t_i = \begin{cases} 1, & \Phi_\zeta(x_i) \in \mathcal{M} \\ 0, & \text{otherwise} \end{cases}, \quad s_i = \begin{cases} 1, & \Phi_\zeta(g_\theta(z_i)) \in \mathcal{M} \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

This is because the SVM cost function value is not dependent on the feature vectors outside of the margin boundaries and is now fully determined by the supporting vectors in $\mathcal{M}$.

Accordingly, the discriminator update is given by following SGD updates:

$$\zeta \quad \leftarrow \quad \zeta + \eta\sum_{i=1}^{n}\langle w^{SVM}, t_i\nabla_\zeta\Phi_\zeta(x_i) - s_i\nabla_\zeta\Phi_\zeta(g_\theta(z_i))/n\rangle \tag{20}$$

In another word, to update the discriminator parameters, we only need to push out the supporting vectors toward the margin boundaries.

On the other hand, generator update requires more geometric intuition. As shown in Fig. 3, the generator update tries to move the fake feature vectors toward the normal vector direction of the separating hyperplane so that they can be classified as the true feature vectors. This means that the generator update should be given by the following minimization problem:

$$\min_\theta L_{w,b,\zeta}(\theta)$$

where

$$L_{w,b,\zeta}(\theta) \quad := \quad -\frac{1}{n}\sum_{i=1}^{n} D_{w,b,\zeta}(g_\theta(z_i)) \tag{21}$$

with the linear classifier

$$D_{w,b,\zeta}(x) := \langle w, \Phi_\zeta(x)\rangle + b\ . \tag{22}$$

This is because $-D_{w,b,\zeta}(x)$ gets smaller as $\Phi_\zeta(g_\theta(z))$ moves toward the upper-left side in Fig. 3 along the normal vector direction w^{SVM}. This results in the following SGD updates:

$$\theta \quad \leftarrow \quad \theta + \eta\sum_{i=1}^{n}\langle w^{SVM}, \nabla_\theta\Phi_\zeta(g_\theta(z_i))\rangle/n\ . \tag{23}$$

Note that the SGD updates (20) and (23) are strikingly similar to (9) and (10) of McGAN. Aside from the different choice of separating hyperplane by (16), the discriminator update (20) has additional geometric scaling factors (t_i, s_i). As will be shown in Appendix A, the appearance of the geometric scaling factors is a recurrent theme in geometric interpretation of GAN and its variants, which we believe is fundamental to account for the geometry of the classifiers.

3.3 Convergence of Geometric GAN

In order to show the convergence of the geometric GAN to a Nash equilibrium, we investigate the behaviour at large sample limit. Specifically, as $n \to \infty$ for a fixed C, the soft margin SVM cost in (14) becomes

$$\begin{aligned} R(D,g) &= E_{x\sim P_x}\left[\max\left(0, 1 - D_{w,b,\zeta}(x)\right)\right] + E_{z\sim P_z}\left[\max\left(0, 1 + D_{w,b,\zeta}(g_\theta(z))\right)\right] \\ &= \int \left[1 - D_{w,b,\zeta}(x)\right]_+ dP_x + \int \left[1 + D_{w,b,\zeta}(g_\theta(z))\right]_+ dP_z \\ &= \int p_x(x)\left[1 - D_{w,b,\zeta}(x)\right]_+ dx + \int p_{g_\theta}(x)\left[1 + D_{w,b,\zeta}(x)\right]_+ dx \end{aligned}$$

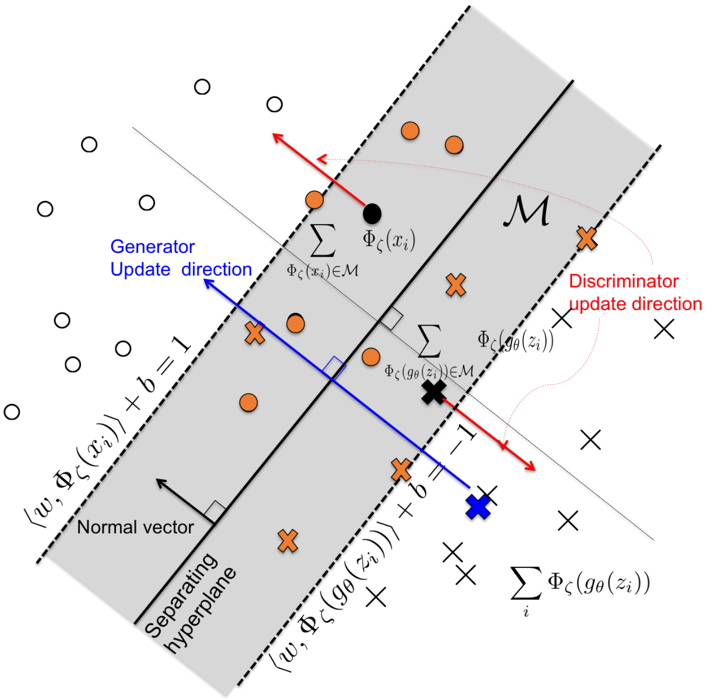

Generator
Update direction
Discriminator
update direction
$\mathcal{M}$
$\sum_{\Phi_\zeta(x_i)\in\mathcal{M}}$
$\Phi_\zeta(x_i)$
$\sum_{\Phi_\zeta(g_\theta(z_i))\in\mathcal{M}}$
$\Phi_\zeta(g_\theta(z_i))$
$\langle w, \Phi_\zeta(x_i)\rangle + b = 1$
$\langle w, \Phi_\zeta(g_\theta(z_i))\rangle + b = -1$
Normal vector
Separating
hyperplane
$\sum_i \Phi_\zeta(g_\theta(z_i))$

Figure 3: Geometric GAN using SVM hyperplane. Discriminator and generator update directions are shown.

where $[x]_+ = \max\{0, x\}$ and $D_{w,b,\zeta}(x)$ is a linear discriminator in (22) parameterized by (w, b, ζ). Here, $p_x(x)$ and $p_{g_\theta}(x)$ denote the probability density functions (pdf) for the distribution P_x and $P_z(g_\theta(z))$, respectively. Similarly, the generator cost function in (21) becomes

$$\begin{aligned} L(D,g) &= -E_{z\sim P_z}\left[D_{w,b,\zeta}(g_\theta(z))\right] \\ &= -\int p_{g_\theta}(x) D_{w,b,\zeta}(x)dx \end{aligned}$$

Then, the adversarial training between discriminator and generator can be achieved by the following alternating minimization:

$$\min_D R(D,g) := \min_{w,b,\zeta} R(D_{w,b,\zeta}, g) \tag{24}$$

$$\min_g L(D,g) := \min_\theta L(D, g_\theta) \tag{25}$$

Suppose that the optimal solution of the aforementioned adversarial training is a pair (D^*, g^*). Then, we can prove the following key convergence result.

Theorem 3.1. *Suppose that* (D^*, g^*) *is a minimizer of the alternating minimization of* (24) *and* (25)*. Then,* $p_{g*}(x) = p_x(x)$ *almost everywhere, and* $R(D^*, G^*) = 2$.

Proof. See Appendix B. □

In the following example, we provide a specific example where the discriminator and generator cost function has close form expressions, which also has intuitive meaning of the minimum value 2 in Theorem 3.1. In particular, we consider an example of learning parallel lines as in the original Wasserstein GAN paper [7].

Example 3.1 (Learning parallel lines)**.** *Let* $u \sim U[0,1]$ *denote the uniform distribution on the unit interval. Let* P_x *be the distribution of* $x = (0, u) \in \mathbb{R}^2$*, uniform on a straight vertical line passing through the origin. Suppose that the generator sample is given by* $g_\theta(z) = (\theta, z)$ *with* θ *a single real parameter. We can easily see that the SVM separating hyperplane is given by*

$$\langle w, x\rangle + b = 0$$

where $w = (-1, 0), b = \theta/2$ *if* $\theta \geq 0$ *and* $w = (1, 0), b = -\theta/2$ *if* $\theta < 0$*. Thus, the generator cost function becomes*

$$\begin{aligned} L(D, g) &= -E\left[\langle w, g_\theta(z)\rangle + b\right] \\ &= |\theta|/2 \end{aligned}$$

which achieves its minimum at $\theta^* = 0$*. Then, the corresponding discriminator cost value at* $n \to \infty$ *is given by*

$$\begin{aligned} R(D^*, g^*) &= \lim_{\theta \to 0} \left\{E\left[1 - \langle w, x\rangle - b\right]_+ + E\left[1 + \langle w, g_\theta(z)\rangle + b\right]_+\right\} \\ &= \lim_{\theta \to 0} 2\left[1 - |\theta|/2\right]_+ \\ &= 2 \end{aligned}$$

which coincides the results by Theorem 3.1.

In this example, $R(D^*, g^*) = 2$ because all the true and fake samples lies on the separating hyperplane. This informs that at the Nash equilibrium of this problem, all the true samples and the fake samples are not separable, which is the desired property of GAN training. However, Theorem 3.1 is only a necessary condition to make the true and fakes sample non-separable. The proof for the sufficiency condition would be very interesting, which is beyond the scope of current paper.

4 Experimental Results

4.1 Mixture of Gaussians

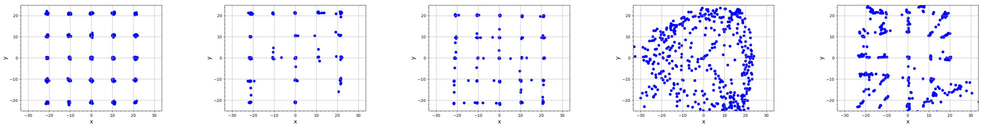
In order to evaluate the proposed geometric GAN, we perform comparative studies with three representative types of GANs; 1) Jenson-Shannon (GAN) [1], 2) mean difference in l_∞ (Wasserstein GAN) [7], and 3) mean difference in l_2 [9]. Here, the behavior of the maximum margin separating hyperplane of the geometric GAN is empirically analyzed against those of the aforementioned approaches.

In addition, to evaluate the dependency of each variants on Lipschitz continuity constraints, the Lipschitz constraints suggested in [7, 18] was also applied to each adversarial training approach. More specifically, the parameters (w, b) of the final linear layer in discriminator is determined to represent the aforementioned hyperplane properties, whereas the Lipschitz constraints are applied for other network parameters, such as ζ in $\Phi_\zeta(x)$ and θ in $g_\theta(z)$. In this paper, we only consider Lipschitz density constraints in [18], so we follow to use weight decay on generators $g_\theta(x)$ and feature space mapping $\phi_\zeta(x)$ in discriminators.

We test the four hyperplane searching approaches for discriminators, as well as their complementary generator losses, on two dimensional synthetic data. The synthetic data consists of 100K data points generated from a mixture of 25 Gaussians, akin to the data that have been used for describing mode collapsing behaviors of GANs [19, 20, 21]. Specifically, the means of the Gaussians are evenly spaced as a 5 by 5 grid along x and y axis from -21 to 21. The standard deviation of each normal distribution is 0.316 (so that the variance would be 0.1). The sampled data from the true distribution can be seen in Figure 4 and 5.

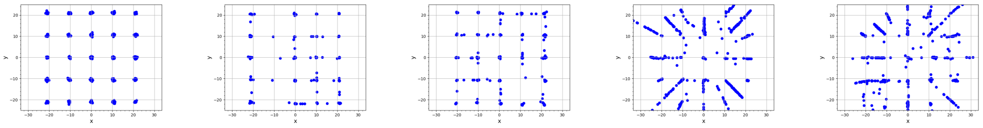
For discriminator and generator, a multi-layered fully-connected neural network architecture is used, as described below. RMSprop [22] is used to train these networks, except vanilla GAN (without any Lipschitz constraints). For vanilla GAN, Adam [23] with momentum $\beta_1 = 0.5$ is used. Base learning rate is set to 0.001. When weight clipping is applied, parameters in feature mapping $\phi_\zeta(x)$ is clipped within the range of $[-0.01, 0.01]$. When weight projection on unit l_2 norm is applied, the following rule, $p = \min\{1, 1/\|p\|_2\} \times p$ described in [9] is used to update any parameter p for every iteration. For weight decay, weight decaying parameter is set to 0.001. Batch size is set to 500 for all experiment. For the number of discriminator update K_d and the one of generator update K_g, we set them as 1, i.e. $(K_d = 1, K_g = 1)$.

- **Discriminator:** $\texttt{FC}(2, 128)$-$\texttt{ReLU}$-$\texttt{FC}(128, 128)$-$\texttt{ReLU}$-$\texttt{FC}(128, 128)$-$\texttt{ReLU}$-$\texttt{FC}(128, 1)$
- **Generator:** $\texttt{FC}(4, 128)$-$\texttt{BN}$-$\texttt{ReLU}$-$\texttt{FC}(128, 128)$-$\texttt{BN}$-$\texttt{ReLU}$-$\texttt{FC}(128, 128)$-$\texttt{BN}$-$\texttt{ReLU}$-$\texttt{FC}(128, 2)$



(a) True data (b) GAN [1] (c) Proposed (d) WGAN [7] (e) meanGAN + wproj on ϕ_ζ [9]

Figure 4: Generated samples of GAN variants for the mixture of 25 Gaussians, and the ones from the true data distribution.



(a) True data (b) GAN + wdecay on Φ_ζ and g_θ (c) Proposed + wdecay on Φ_ζ and g_θ (d) WGAN + wdecay on Φ_ζ and g_θ (e) meanGAN + wdecay on Φ_ζ and g_θ

Figure 5: Generated samples of GAN variants under trained with Lipschitz density constraints suggested in [18] for the mixture of 25 Gaussians, and the sample from the true data distribution. During training, weight decay was applied on ζ in $\Phi_\zeta(x)$ and θ in $g_\theta(z)$ in addition to their own hyperplane constraints.

The results of the experiment with the mixture of 25 Gaussians are illustrated in Figure 4 and 5. Amongst all GAN variants in this experiment, geometric GAN demonstrated the least mode collapsing behavior independently with Lipschitz continuity regularization constraints.

As shown in Fig. 5, under the same Lipschitz density constraints, linear hyperplane approaches demonstrated less mode collapsing behaviors by virtue of consistent gradients unlike nonlinear separating hyperplane of original GAN. However, mean difference-driven hyperplanes in Wasserstein GAN or McGAN led generators to the mean of arbitrary number of modes in true distributions since the characteristics of mean difference. One the other hand, geometric GAN generally showed robust and consistent convergence behavior towards true distributions.

4.2 Image Datasets

In order to analyze the proposed method on large-scale dataset, the geometric GAN is empirically analyzed on well-studied datasets in the context of adversarial training; MNIST, CelebA, and LSUN datasets. Since consistent quantitative measures are still under debate, we only perform qualitative comparisons of generated samples from the learned generators of the propsed method against the results of previous literatures. In favor of fair comparisons with other adversarial training methods, we adopt the settings from the previous literatures except the hyperparameters of stochastic optimizations and the tuning parameter of the proposed method.

The DCGAN neural network architectures [24] was used, including batch normalization for generator. Note that the currently known adversarial training methods that demonstrated stable learning without batch normalization [7, 9, 18] have resorted to Lipschitz constraints; therefore, it can also be applied to other adversarial training criterions, including geometric GAN, in order to train batch normalization-free generators.

Each pixel value in input image was rescaled to $[-1, 1]$ for all dataset, including MNIST dataset. During all training, mini-batch size was set to 64. For stochastic gradient update during training, RMSprop was used [22]. The number of generator's updates per each discriminator's update is set to 10 ($K_d = 1, K_g = 10$). Learning rate is set to 0.0002 for both discriminators and generators, and a tuning parameter C for discriminator is set to 1.

Specifically for MNIST dataset, input images were resized to 64 by 64 pixels in order to use the same DCGAN network architecture, and the number of epochs for training was set to 20. For CelebA dataset, input images were resized to 96 by 96 pixels and center-cropped with 64 by 64 pixels, and the number of epochs for training is set to 50. For LSUN dataset, only bedroom dataset is used, and an input image is resized to 64 by 64 pixels. The number of epochs for training is set to 2 for LSUN dataset.

The results in Figure 6, 7, and 8 clearly show that the geometric GAN generates very realistic images without mode collapsing or divergent behaviours.

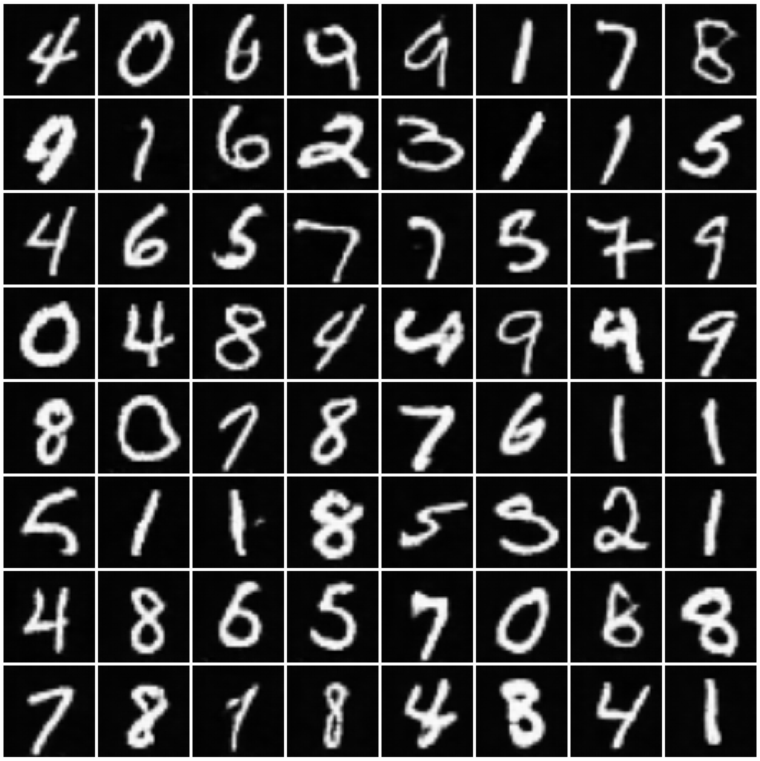


Figure 6: Generated samples of Geometric GAN trained for MNIST dataset.

5 Conclusion

This paper proposed a novel geometric GAN using SVM separating hyperplane, based on geometric intuitions revealed from previous adversarial training approaches. The geometric GAN was based on SVM separating hyperplanes that has the maximal margins between the two classes. Compared to the most of the existing approaches that are based on statistical design criterion, the geometric GAN is derived based on geometric intuition similar to the derivation of SVM. Extensive numerical experiments showed that the proposed method has demonstrated less mode collapsing and more stable training behavior. Moreover, our theoretical results showed that the proposed algorithm converges to the Nash equilibrium between the discriminator and generator, which has also geometric meaning.

Acknowledgement

This work is supported by Korea Science and Engineering Foundation, Grant number NRF2016R1A2B3008104. The first author would like to thank Yunhun Jang for helpful discussions.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

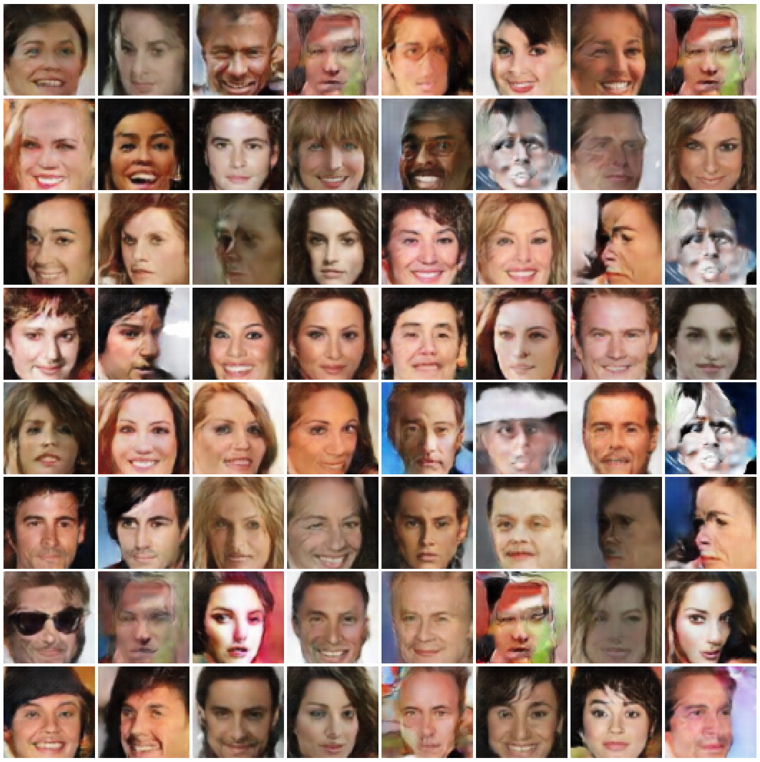


Figure 7: Generated samples of Geometric GAN trained for CelebA dataset.

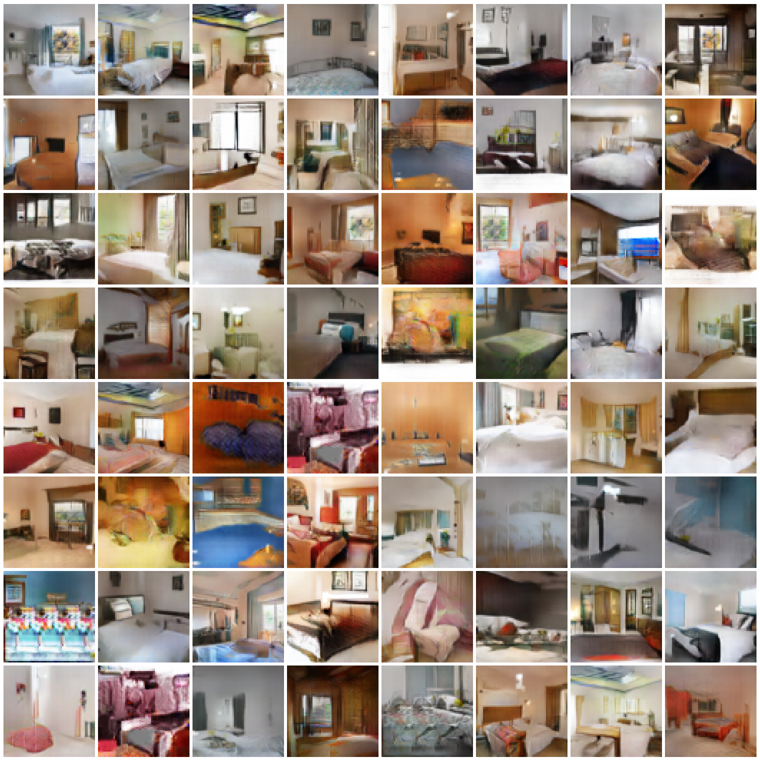


Figure 8: Generated samples of Geometric GAN trained for LSUN dataset.

[2] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[3] S. Nowozin, B. Cseke, and R. Tomioka, “f-GAN: Training generative neural samplers using variational divergence minimization,” in *Advances in Neural Information Processing Systems*, 2016, pp. 271–279.

[4] Y. Li, K. Swersky, and R. Zemel, “Generative moment matching networks,” in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 1718–1727.

[5] G. Dziugaite, D. Roy, and Z. Ghahramani, “Training generative neural networks via maximum mean discrepancy optimization,” in *Uncertainty in Artificial Intelligence-Proceedings of the 31st Conference, UAI 2015*, 2015, pp. 258–267.

[6] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” in *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, vol. 2016, 2017.

[7] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” *arXiv preprint arXiv:1701.07875*, 2017.

[8] A. Müller, “Integral probability metrics and their generating classes of functions,” *Advances in Applied Probability*, vol. 29, no. 02, pp. 429–443, 1997.

[9] Y. Mroueh, T. Sercu, and V. Goel, “McGan: Mean and covariance feature matching GAN,” *arXiv preprint arXiv:1702.08398*, 2017.

[10] J. S. Marron, M. J. Todd, and J. Ahn, “Distance-weighted discrimination,” *Journal of the American Statistical Association*, vol. 102, no. 480, pp. 1267–1271, 2007.

[11] I. Carmichael and J. Marron, “Geometric insights into support vector machine behavior using the KKT conditions,” *arXiv preprint arXiv:1704.00767*, 2017.

[12] J. Ahn and J. Marron, “The maximal data piling direction for discrimination,” *Biometrika*, vol. 97, no. 1, pp. 254–259, 2010.

[13] J. Zhao, M. Mathieu, and Y. LeCun, “Energy-based generative adversarial network,” *arXiv preprint arXiv:1609.03126*, 2017.

[14] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

[15] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, “Hilbert space embeddings and metrics on probability measures,” *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1517–1561, 2010.

[16] B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. R. Lanckriet, “On integral probability metrics,ϕ-divergences and binary classification,” *arXiv preprint arXiv:0901.2698*, 2009.

[17] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, “The entire regularization path for the support vector machine,” *Journal of Machine Learning Research*, vol. 5, no. Oct, pp. 1391–1415, 2004.

[18] G.-J. Qi, “Loss-sensitive generative adversarial networks on lipschitz densities,” *arXiv preprint arXiv:1701.06264*, 2017.

[19] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. C. Courville, “Adversarially learned inference,” *arXiv preprint arXiv:1606.00704*, 2016.

[20] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, “Unrolled generative adversarial networks,” *arXiv preprint arXiv:1611.02163*, 2016.

[21] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, “Mode regularized generative adversarial networks,” *arXiv preprint arXiv:1612.02136*, 2016.

[22] T. Tieleman and G. Hinton, “RMSprop Gradient Optimization,” *http://www.cs.toronto.edu/tijmen/csc321/slides/lecture-slides-lec6.pdf.*

[23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[24] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

A Geometric interpretation of GAN and its variants

This appendix provides geometric interpretation of GAN and its variants. In particular, we consider a specific form of the discriminator $D_{w,\zeta}(x_i)$ given by

$$D_{w,\zeta}(x_i) = S_f(V_{w,\zeta}(x_i)), \quad \text{where} \quad V_{w,\zeta}(x_i) := \langle w, \Phi_\zeta(x_i)\rangle \tag{26}$$

where S_f is an output activation function, $V_{w,\zeta}(x_i)$ is the output layer composed of linear layer w and the convolutional neural network below corresponding to $\Phi_\zeta(x)$. Under this choice of the discriminator, we will show that the differences between existing approaches come from the choice of the separating hyperplanes and geometric scaling factors.

A.1 GAN

Recall that the empirical estimate of the GAN cost in (2) is given by :

$$\hat{L}_{GAN}(w,\zeta,\theta) := \frac{1}{n}\sum_{i=1}^n \log D_{w,\zeta}(x_i) + \frac{1}{n}\sum_{i=1}^n \log(1 - D_{w,\zeta}(g_\theta(z_i)),$$

We now define geometric scaling factors for true and synthetic (or fake) feature vectors:

$$t_i := \frac{S'\left(\langle w, \Phi_\zeta(x_i)\rangle\right)}{D(x_i)}, \qquad s_i := \frac{S'\left(\langle w, \Phi_\zeta(g_\theta(z_i))\rangle\right)}{1 - D(g_\theta(z_i))}$$

In particular, if the activation function is the sigmoid, i.e. $S(u) = 1/(1+e^{-u})$, then we can easily see that

$$t_i := 1 - D(x_i), \qquad s_i := D(g_\theta(z_i)).$$

Then, the separating hyperplane update is given by:

$$w^{GAN} \quad \leftarrow \quad w^{GAN} + \eta \sum_{i=1}^n \left(t_i \Phi_\zeta(x_i) - s_i \Phi_\zeta(g_\theta(x_i))\right)/n \tag{27}$$

Using another application of chain rules,

$$\zeta \quad \leftarrow \quad \zeta + \eta \sum_{i\in I} \langle w^{GAN}, t_i \nabla_\zeta \Phi_\zeta(x_i) - s_i \nabla_\zeta \Phi_\zeta(g_\theta(z_i))\rangle/n \tag{28}$$

$$\theta \quad \leftarrow \quad \theta + \eta \sum_{i\in I} \langle w^{GAN}, s_i \nabla_\theta \Phi_\zeta(g_\theta(z_i))\rangle/n \tag{29}$$

Aside from different choice of separating hyperplane, the only difference is that the features vectors needs to be scaled appropriated using geometric scaling parameters. In fact, the scale parameter is directly related to the geometry of the underlying curved feature spaces due to the $\log(\cdot)$ and nonlinear activations.

From (29), we can easily see that as discriminator becomes accurate, we have $s_i = D\left(g_\theta(z_i)\right) \simeq 0$, so the update of the generator becomes more difficult. This is the main technical limitation of the GAN training.

A.2 f-GAN

The f-GAN formulation is given by the minmax game of the following empirical cost:

$$F(w,\zeta,\theta) \quad = \quad \frac{1}{n}\sum_{i=1}^n S_f(V_{w,\zeta}(x_i)) - \frac{1}{n}\sum_{i=1}^n f^*(S_f(V_{w,\zeta}(g_\theta(z_i)))$$

where f^* is the convex conjugate of the divergence function f. We again define a geometric scale factors for true and fake feature vectors:

$$t_i := S_f'\left(\langle w, \Phi_\zeta(x_i)\rangle\right), \qquad s_i := (f^*)'\left(S_f(\langle w, \Phi_\zeta(x_i)\rangle)\right) S_f'\left(\langle w, \Phi_\zeta(x_i)\rangle\right).$$

The explicit forms of the geometric scaling factors for different f-divergences are‘ summarized in Table 1,

Name	$S_f(v)$	$f^*(v)$	$t_i(u)$	$s_i(u)$
Total variation	$\frac{1}{2}\tanh(v)$	v	$\frac{1}{2}\coth(u)$	$\frac{1}{2}\coth(u)$
Kullback-Leiber (KL)	v	$\exp(v-1)$	1	$\exp(u-1)$
Reverse KL	$-\exp(v)$	$-1-\log(-v)$	$-\exp(u)$	1
Pearson χ^2	v	$v^2/4+v$	1	$u/2+1$
Jensen-Shannon	$\log(2)-\log(1+\exp(-v))$	$-\log(2-\exp(v))$	$\frac{e^{-u}}{1+e^{-u}}$	$\frac{1}{1+e^{-u}}$
GAN	$-\log(1+\exp(-v))$	$-\log(1-\exp(v))$	$\frac{e^{-u}}{1+e^{-u}}$	$\frac{1}{1+e^{-u}}$

Table 1: Recommended final layer activation functions for f-GAN [3] and their geometric scaling factors.

Then, the separating hyperplane update is given by:

$$w^{fGAN} \quad \leftarrow \quad w^{fGAN} + \eta \sum_{i=1}^{n} \left(t_i \Phi_\zeta(x_i) - s_i \Phi_\zeta(g_\theta(x_i))\right)/n \tag{30}$$

From the chain rules, we have discriminator and generator update rules:

$$\zeta \quad \leftarrow \quad \zeta + \eta \sum_{i \in I} \langle w^{fGAN}, t_i \nabla_\zeta \Phi_\zeta(x_i) - s_i \nabla_\zeta \Phi_\zeta(g_\theta(z_i))\rangle/n \tag{31}$$

$$\theta \quad \leftarrow \quad \theta + \eta \sum_{i \in I} \langle w^{fGAN}, s_i \nabla_\theta \Phi_\zeta(g_\theta(z_i))\rangle/n \tag{32}$$

Note that f-GAN is only different from each other in their construction of the weight coefficient (t_i, s_i) (see Table 1) that reflects the underlying geometry of the curved feature space. Other than the total variation-based divergence, the scaling factors are asymmetric. Thus, controlling the balance between discriminator and generator updates are one of the important technical issues of f-GAN training.

A.3 Wasserstein GAN

Wasserstein GAN [7] minimizes the following IPM:

$$d_{\mathcal{F}}(P,Q) = \sup_{\|f\|_L \leq 1} \frac{1}{n}\sum_{i=1}^{n} f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(g_\theta(z_i))$$

where $\|f\|_L := \sup\{|f(x) - f(y)|/\rho(x,y) : x \neq y \in M\}$ is called the Lipschitz seminorm of a real-valued function f on M. Using the discriminator model (26), the Wasserstein GAN update can be written by:

$$\min_{\theta} \max_{\|w\|_\infty \leq 1, \zeta} \quad \left\langle w, \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \Phi_\zeta(x_i) - \tfrac{1}{n}\sum_{i=1}^{n} \Phi_\zeta(g_\theta(z_i))\right\rangle \tag{33}$$

Therefore, other than the mean difference on l_∞ ball for the hyperplane normal vector w update, the W-GAN update is same as the mean matching GAN update with geometric scaling factor $t_i = s_i = 1, \forall i$.

A.4 Energy-based GAN

For a given a positive margin m, the energy-based GAN (EBGAN) is given by the alternating minimization of the discriminator and generator cost functions [13]:

$$L_D(w,\zeta) \quad = \quad \frac{1}{n}\sum_{i=1}^{n}\Big(D_{w,\zeta}(x_i) + [m - D_{w,\zeta}(g_\theta(z_i))]_+\Big)$$

$$L_G(\theta) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} D_{w,\zeta}(g_\theta(z_i))$$

where $[x]_+ = \max\{0, x\}$. For a function $\psi(y) = ay + b[m-y]_+$ with $y, a, b \geq 0$, its subgradient is given by:

$$\psi'(y) = \begin{cases} a-b, & y \in [0, m] \\ a, & y \in (m, \infty) \\ [a-b, a], & y \in m \end{cases}$$

Due to the margin, geometric scale factors for true and fake feature vectors should be defined accordingly. More specifically, we have

$$t_i := S_f'\left(\langle w, \Phi_\zeta(x_i)\rangle\right), \qquad s_i^G := S_f'\left(\langle w, \Phi_\zeta(x_i)\rangle\right), \quad s_i := \begin{cases} s_i, & D_{w,\zeta}(g_\theta(z_i)) \in [0, m] \\ 0, & \text{otherwise} \end{cases}$$

Then, the separating hyperplane update is given by:

$$w^{fGAN} \quad \leftarrow \quad w^{fGAN} + \eta \sum_{i=1}^{n} \left(t_i \Phi_\zeta(x_i) - s_i \Phi_\zeta(g_\theta(x_i))\right)/n \tag{34}$$

Similarly, we have

$$\zeta \quad \leftarrow \quad \zeta + \eta \sum_{i \in I} \langle w^{fGAN}, t_i \nabla_\zeta \Phi_\zeta(x_i) - s_i \nabla_\zeta \Phi_\zeta(g_\theta(z_i))\rangle / n \tag{35}$$

$$\theta \quad \leftarrow \quad \theta + \eta \sum_{i \in I} \langle w^{fGAN}, s_i^G \nabla_\theta \Phi_\zeta(g_\theta(z_i))\rangle / n \tag{36}$$

It is worthy to note that the introduction of *margin* appears similar to our geometric GAN with SVM hyperplane. In particular, when a linear activation function is used, we have $t_i = s_i^G = 1$, the update equations (35) and (36) appears very similar to (20) and (23), respectively. However, there exists fundamental differences. First, in EB-GAN, only the fake samples outside the margins are excluded for the hyperplane and discriminator updates. On the other hand, in geometric GAN, both the true and fake samples outside the margins are excluded for the hyperplane and discriminator updates. The symmetric exclusion in geometric GAN is observed to make the algorithm more robust to outliers. Second, in EBGAN, the margin is defined for the discriminator values. On the other hand, in geometric GAN, the margin is determined by the geometric distance between the feature vectors. Therefore, it is much easier to rely on geometric intuition in designing the geometric GAN.

A.5 Empirical risk minimization

The empirical risk minimization (ERM) with l_2 cost is one of the standard method for regression problems. Although the empirical risk minimization (ERM) is rarely used for generator model, our analysis also provides the geometric intuition of ERM update.

Specifically, for a given mini-batch training data set $S = \{(z_1, x_1), \cdots, (z_n, x_n)\}$, recall that the empirical risk minimization (ERM) in the feature space [14] is given by

$$\min_\theta \frac{1}{2} \sum_{i=1}^{n} \|\Phi_\zeta(x_i) - \Phi_\zeta(g_\theta(z_i))\|^2 \tag{37}$$

Then, the stochastic gradient for (37) can be represented in the identical form to (7):

$$\zeta \quad \leftarrow \quad \zeta + \eta \sum_{i \in I} \langle w_i^{ERM}, \nabla_\zeta \Phi_\zeta(x_i) - \nabla_\zeta \Phi_\zeta(g_\theta(z_i))\rangle / n \tag{38}$$

$$\theta \quad \leftarrow \quad \theta + \eta \sum_{i=1}^{n} \left\langle w_i^{ERM}, \nabla_\theta \Phi_\zeta(g_\theta(z_i))\right\rangle / n \tag{39}$$

where w_i^{ERM} is now defined as

$$w_i^{ERM} \quad = \quad \Phi_\zeta(x_i) - \Phi_\zeta(g_\theta(z_i)) \tag{40}$$

which is dependent on the sample index i. Therefore, aside from the geometric scaling factors, the main difference comes from the separating hyperplane for linear classifiers. More specifically, the hyperplane for geometric GAN is obtained for samples within each mini-batch, while the classifier for regression is optimally designed for each pair of samples.

B Proof for Theorem 3.1

The proof technique is inspired from that of EB-GAN [13]. We first need the following two lemmas as the extensions of Lemma 1 in [13].

Lemma B.1. *Let $\varphi(y) = (m - y) + [m + y]_+$. The minimum of $\varphi(y)$ is $2m$ and is reached at all $y \geq -m$.*

Proof. If $y \geq -m$, then $\varphi(y) = 2m$. If $y \leq -m$, the $\varphi(y) = -2y$, whose minimum $2m$ is achieved at $y = -m$. □

Lemma B.2. *For given $\alpha, \beta \geq 0$, The minimum of $\varphi(y) = \alpha\,[m - y]_+ + \beta\,[m + y]_+$ exist if $y \in [-m, m]$. More specifically, the minimum of $\varphi(y)$ is $2\beta m$ at $y = m$ if $\alpha > \beta$, or $2\alpha m$ at $y = -m$ if $\alpha \leq \beta$.*

Proof. If $y \geq m$, then $\varphi(y) = \beta\,[m + y]_+ = \beta(m + y) \geq 2\beta m$. Thus, $\inf_{y\in[m,\infty)} \varphi(y) = 2\beta m$ at $y = m$. Similarly, if $y \leq -m$, then $\inf_{y\in(-\infty,-m]} \varphi(y) = 2\alpha m$ at $y = -m$. For $y \in [-m, m]$, $\varphi(y) = \alpha(m-y)+\beta(m+y) = (\alpha+\beta)m+(\beta-\alpha)y$. If $\alpha > \beta$, $\inf_{y\in[-m,m]} \varphi(y) = 2\beta m$ at $y = m$ since $\varphi(y)$ is a decreasing function on $y \in [-m, m]$. Similarly, if $\alpha \leq \beta$, $\inf_{y\in[-m,m]} \varphi(y) = 2\alpha m$ at $y = -m$ since $\varphi(y)$ is increasing at $y \in [-m, m]$. □

Now we are ready for the proof.

Proof of Theorem 3.1. Since $R(D, g)$ and $L(D, g)$ are lower semi-continuous functions, $R(D^*, g^*)$ has a finite value for the optimal solution (D^*, g^*). Moreover, due to the alternating minimization, the pair satisfies:

$$R(D^*, g^*) \leq R(D, g^*), \quad \forall D \tag{41}$$

$$L(D^*, g^*) \leq L(D^*, g), \quad \forall g \tag{42}$$

First, define the set

$$A = \{x | p_x(x) \leq p_{g^*}(x)\}$$

and observe that

$$\begin{aligned} R(D, g^*) &= \int p_x(x)\big[1 - D_{w,b,\zeta}(x)\big]_+ + p_{g^*_\theta}(x)\big[1 + D_{w,b,\zeta}(x)\big]_+ dx \\ &= \int \mathbb{1}_A(x)p_x(x)\big[1 - D_{w,b,\zeta}(x)\big]_+ dx \\ &\quad + \int \mathbb{1}_A(x)p_{g^*_\theta}(x)\big[1 + D_{w,b,\zeta}(x)\big]_+ dx \\ &\quad + \int \mathbb{1}_{A^c}(x)p_x(x)\big[1 - D_{w,b,\zeta}(x)\big]_+ dx \\ &\quad + \int \mathbb{1}_{A^c}(x)p_{g^*_\theta}(x)\big[1 + D_{w,b,\zeta}(x)\big]_+ dx \end{aligned}$$

where A^c denotes the complementary set and $\mathbb{1}_A(x)$ is an indicator function

$$\mathbb{1}_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}.$$

From Lemma B.2, we know that 1) when $p_x(x) < p_{g^*_\theta}(x)$, the term within the integral achieves its minimum value of $2p_x(x)$ at $D^*(x) = -1$, or 2) when $p_x(x) > p_{g^*_\theta}(x)$, the term within the integral achieves its minimum value of $2p_{g^*_\theta}(x)$ at $D^*(x) = 1$. Therefore,

$$\begin{aligned} R(D^*, g^*) &= 2\int \mathbb{1}_A(x)p_x(x)dx + 2\int \mathbb{1}_{A^c}(x)p_{g^*_\theta}(x)dx \\ &= 2\int \big[\mathbb{1}_A(x)p_x(x) + \big(1 - \mathbb{1}_A(x)\big)p_{g^*_\theta}(x)\big]dx \\ &= 2 + 2\int \mathbb{1}_A(x)\big(p_x(x) - p_{g^*_\theta}(x)\big)dx \\ &\leq 2 \end{aligned} \tag{43}$$

where the last inequality comes from $p_x(x) - p_{g_\theta}(x) \leq 0$ for all $x \in A$.

Second, we will show that $R(D^*, g^*) \geq 2$. Because (42) holds for arbitrary pdf $p(x)$, we have

$$\int p_{g_\theta^*}(x)\left(-D^*(x)\right)dx \quad \leq \quad \int p_x(x)\left(-D^*(x)\right)dx$$

By adding 1 to both sides, we have

$$\begin{aligned} \int p_{g_\theta^*}(x)\left(1 - D^*(x)\right)dx \quad &\leq \quad \int p_x(x)\left(1 - D^*(x)\right)dx \\ &\leq \quad \int p_x(x)\left[1 - D^*(x)\right]_+ dx \end{aligned}$$

where the last inequality comes from $x \leq [x]_+ = \max\{0, x\}$. Now, by adding $\int p_{g_\theta^*}(x)\big[1 + D^*(x)\big]_+ dx$ on both sides, we have:

$$\begin{gathered} \int p_{g_\theta^*}(x)\big(1 - D^*(x)\big)dx + \int p_{g_\theta^*}(x)\big[1 + D^*(x)\big]_+ dx \\ \leq \int p_x(x)\big[1 - D^*(x)\big]_+ dx + \int p_{g_\theta^*}(x)\big[1 + D^*(x)\big]_+ dx = R(D^*, g^*) \end{gathered} \tag{44}$$

From Lemma B.1, we know that $\big(1 - D^*(x)\big) + \big[1 + D^*(x)\big]_+ \geq 2$. Thus, we have

$$\begin{aligned} R(D^*, g^*) \quad &\geq \quad \int p_{g_\theta^*}(x)\big(1 - D^*(x)\big)dx + \int p_{g_\theta^*}(x)\big[1 + D^*(x)\big]_+ dx \\ &\geq \quad \int p_{g_\theta^*}(x) 2 dx \quad \geq \quad 2 \end{aligned}$$

Thus,

$$2 \leq R(D^*, g^*) \leq 2 \tag{45}$$

Finally, the equality in (43) holds if and only if

$$\int \mathbb{1}_A(x)\big(p_x(x) - p_{g_\theta^*}(x)\big)dx = 0$$

The above equalities hold if and only if $p_{g*}(x) = p_{data}(x)$ almost everywhere [13]. This concludes the proof.

□