UNDERSTANDING THE EFFECTIVENESS OF LIPSCHITZ-CONTINUITY IN GENERATIVE ADVERSARIAL NETS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we investigate the underlying factor that leads to the failure and success in training of GANs. Specifically, we study the property of the optimal discriminative function $f^*(x)$ and show that $f^*(x)$ in most GANs can only reflect the local densities at x, which means the value of $f^*(x)$ for points in the fake distribution (P_g) does not contain any information useful about the location of other points in the real distribution (P_r). Given that the supports of the real and fake distributions are usually disjoint, we argue that such a $f^*(x)$ and its gradient tell nothing about "how to pull P_g to P_r", which turns out to be the fundamental cause of failure in training of GANs. We further demonstrate that a well-defined distance metric (including Wasserstein distance) does not necessarily ensure the convergence of GANs. Finally, we propose Lipschitz-continuity condition as a general solution and show that in a large family of GAN objectives, Lipschitz condition is capable of connecting P_g and P_r through $f^*(x)$ such that the gradient $\nabla_x f^*(x)$ at each sample $x \sim P_g$ points towards some real sample $y \sim P_r$.

1 INTRODUCTION

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), as a new way of learning generative models, have recently shown promising results in various challenging tasks. Although GANs are popular and widely-used (Isola et al., 2016; Brock et al., 2016; Nguyen et al., 2016; Zhu et al., 2017; Karras et al., 2017), they are notoriously hard to train (Goodfellow, 2016). The underlying obstacles, though have been widely studied (Arjovsky & Bottou, 2017; Lucic et al., 2017; Heusel et al., 2017a; Mescheder et al., 2017; 2018), are still not fully understood. In this paper, we study the convergence of GANs from the perspective of the optimal discriminative function $f^*(x)$.

We show that in original GAN and its most variants, $f^*(x)$ is a function of densities at the current point x but does not reflect any information about the densities/locations of other points in the real and fake distributions. Moreover, Arjovsky & Bottou (2017) states that the supports of real and fake distributions are usually disjoint. In this paper, we argue that the fundamental cause of failure in training of GANs (Section 2.1) stems from the combination of the above two facts. The generator uses $\nabla_x f^*(x)$ as the guidance for updating the generated samples, but $\nabla_x f^*(x)$ actually tells nothing about where P_r is. Therefore, the generator is not guaranteed to converge to the case $P_g = P_r$.

Accordingly, Arjovsky et al. (2017) proposed the Wasserstein distance (in its dual form) as an alternative objective, which can properly measure the distance between two distributions no matter whether their supports are disjoint. However, as shown in Section 2.3, when the supports of the P_g and P_r are disjoint, the gradient of $f^*(x)$ in Wasserstein distance also does not reflect any useful information about other points in P_r. Thus it is also unable to guarantee the convergence. Based on this observation, we provide further investigation in Section 2.4 and argue that measuring the distance properly does not necessarily imply that the gradient is well-defined in terms of $\nabla_x f^*(x)$.

In Section 3, we propose incorporating Lipschitz-continuity condition in the objectives of GANs as a general solution, and prove that in a **broad** family of discriminator objectives, Lipschitz-continuity condition can build strong connections between P_g and P_r through $f^*(x)$ such that when the supports of P_r and P_r are disjoint, $\nabla_x f^*(x)$ at each sample $x \sim P_g$ will point towards some real sample $y \sim P_r$. This guarantees that P_g is moving towards P_r at every step.

We extend our discussion on $f^*(x)$ and $\nabla_x f^*(x)$ to the case where the supports of P_g and P_r are overlapped, and present the general arguments in Section 4.1. Subsequently, in Section 4.2, we show that the locality of $f^*(x)$ and $\nabla_x f^*(x)$ in traditional GANs turns out to be an intrinsic cause to mode collapse. Finally, in Section 4.3, we explain the reason of empirical success of traditional GANs under the circumstance that they have no convergence guarantee.

Table 1: Comparison of different objectives in GAN models.

	ϕ	φ	$\mathcal{F}$	$f^*(x)$
JS-Divergence	$-\log(\sigma(-x))$	$-\log(\sigma(x))$	$\{f:\mathbb{R}^n\to\mathbb{R}\}$	$\log\frac{P_r(x)}{P_g(x)}$
Least Square	$(x-\alpha)^2$	$(x-\beta)^2$	$\{f:\mathbb{R}^n\to\mathbb{R}\}$	$\frac{\alpha\cdot P_g(x)+\beta\cdot P_r(x)}{P_g(x)+P_r(x)}$
Wasserstein-1 with Lip_1	x	$-x$	$\{f:\mathbb{R}^n\to\mathbb{R},\ \|f\|_{lip}\le 1\}$	N/A
μ-Fisher IPM	x	$-x$	$\{f:\mathbb{R}^n\to\mathbb{R},\ \mathbb{E}_{x\sim\mu}\|f(x)\|^2\le 1\}$	$\frac{1}{\mathcal{F}_\mu(P_r,P_g)}\frac{P_r(x)-P_g(x)}{\mu(x)}$

2 The Fundamental Cause of Failure in Training of GANs

Typically, the objectives of GANs can be formulated as follows:

$$\begin{aligned}&\min_{f\in\mathcal{F}}\ J_D \triangleq \mathbb{E}_{z\sim P_z}[\phi(f(g(z)))]+\mathbb{E}_{x\sim P_r}[\varphi(f(x))],\\&\min_{g\in\mathcal{G}}\ J_G \triangleq \mathbb{E}_{z\sim P_z}[\psi(f(g(z)))],\end{aligned}\tag{1}$$

where P_z is the source distribution of the generator (usually a Gaussian distribution) in $\mathbb{R}^m$ and P_r is the target (real) distribution in $\mathbb{R}^n$. The generative function $g:\mathbb{R}^m\to\mathbb{R}^n$ learns to output samples that shares the same dimension as P_r, while the discriminative function $f:\mathbb{R}^n\to\mathbb{R}$ learns to output a score indicating the authenticity of a given sample. We denote the implicit distribution of the generated samples as P_g, i.e., $P_g=g(P_z)$.

$\mathcal{F}$ and $\mathcal{G}$ denote discriminative and generative function spaces parameterized by neural networks, respectively; functions ϕ, φ, ψ: $\mathbb{R}\to\mathbb{R}$ are loss metrics. We list the choices of $\mathcal{F}$, ϕ and φ in some representative GAN models in Table 1, where we denote $f^*=\arg\min_{f\in\mathcal{F}} J_D$.

In these GANs, the gradient that the generator receives from the discriminator with respect to a generated sample $x\sim P_g$ is

$$\nabla_x J_G(x)=\nabla_{f(x)}\psi(f(x))\cdot\nabla_x f(x).\tag{2}$$

In Eq. (2), the first term $\nabla_{f(x)}\psi(f(x))$ is a step-related scalar that is out of the scope of our discussion in this paper; the second term $\nabla_x f(x)$ is a vector indicating the direction that the generator should follow for optimizing on sample x.

2.1 $\nabla_x f^*(x)$ on P_g does not reflect useful information about P_r

In this section, we will show that when the supports of P_g and P_r are disjoint, $\nabla_x f^*(x)$ in traditional GANs does not reflect any useful information about P_r, and P_g is not guaranteed to converge to P_r. We argue that this is the fundamental cause of non-convergence and instability in traditional GANs.[1]

2.1.1 The original GAN and Least-Squares GAN

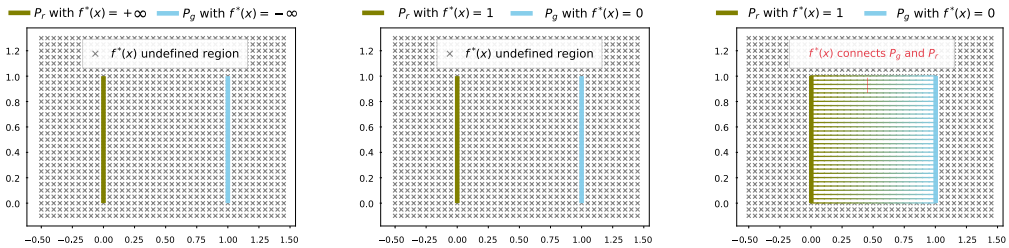
In the simplest case of Eq. (1), e.g., the original GAN (Goodfellow et al., 2014) and Least-Squares GAN (Mao et al., 2016), there is no restriction on $\mathcal{F}$. Therefore, $f^*(x)$ for each point x is independent of other points, and we have

$$f^*(x)=\arg\min_{f(x)\in\mathbb{R}}\ P_g(x)\cdot\phi(f(x))+P_r(x)\cdot\varphi(f(x)),\forall x.\tag{3}$$

Since we assume supports of P_g and P_r are disjoint, we further have

$$f^*(x)=\begin{cases}\arg\min_{f(x)\in\mathbb{R}}\ P_g(x)\cdot\phi(f(x)),\forall x\sim P_g,\\ \arg\min_{f(x)\in\mathbb{R}}\ P_r(x)\cdot\varphi(f(x)),\forall x\sim P_r.\end{cases}\tag{4}$$

[1] In this paper, traditional GANs mainly refers to the original GAN and Least-Squares GAN, where $f^*(x)$ depends only on the densities $P_g(x)$ and $P_r(x)$. Broadly, it refers to all GANs where $f^*(x)$ does not reflect information about the locations of the other points in P_g and P_r, such as the Fisher GAN.



(a) Original GAN (b) Wasserstein distance (c) Lipschitz constraint

Figure 1: In traditional GANs, $f^*(x)$ is only defined on the supports of P_g and P_r and its values do not reflect any information about the locations of other points in P_g and P_r. Therefore, they have no guarantee on the convergence. Wasserstein distance suffers from the same problem. GANs under Lipschitz constraint builds connection between P_g and P_r, where $\nabla_x f^*(x)$ pulls P_g towards P_r.

For $x \sim P_g$, the value of $f^*(x)$ is irrelevant to P_r. Since P_g and P_r are disjoint[2], $\nabla_x f^*(x)$ for $x \sim P_g$ also tells nothing about P_r. In consequence, the generator can hardly learn useful information and is not guaranteed to converge to the case where $P_g = P_r$.

2.1.2 The Fisher GAN

Mroueh et al. (2017) prove that the optimal f^* of μ-Fisher IPM $\mathcal{F}_\mu(P_r, P_g)$, the objective used in Fisher GAN (Mroueh & Sercu, 2017), has the following form

$$f^*(x) = \frac{1}{\mathcal{F}_\mu(P_r, P_g)} \frac{P_r(x) - P_g(x)}{\mu(x)}. \tag{5}$$

where μ is a distribution whose support covers P_r and P_g. Given P_r and P_g are disjoint, we have

$$f^*(x) = \begin{cases} \frac{1}{\mathcal{F}_\mu(P_r,P_g)} \frac{-P_g(x)}{\mu(x)}, & \forall x \sim P_g; \\ \frac{1}{\mathcal{F}_\mu(P_r,P_g)} \frac{P_r(x)}{\mu(x)}, & \forall x \sim P_r; \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Note that the scalar $\frac{1}{\mathcal{F}_\mu(P_r,P_g)}$ is a constant. Eq. (6) also defines $f^*(x)$ on P_g and P_r independently. Therefore, for $x \sim P_g$, $f^*(x)$ and $\nabla_x f^*(x)$ tell nothing about P_r.

2.2 Connection to Gradient Vanishing

The non-convergence problem of the original GAN has once been considered as the gradient vanishing problem. In (Goodfellow et al., 2014), it is addressed by using an alternative objective for the generator. However, it actually only changes the scalar $\nabla_{f(x)}\psi(f(x))$ while the aforementioned problem in $\nabla_x f^*(x)$ still exists. The least-squares GAN (Mao et al., 2016) is proposed to address the gradient vanishing problem, but it also focuses on $\nabla_{f(x)}\psi(f(x))$ basically. As we have discussed, the least-squares GAN also belongs to traditional GANs, which is not guaranteed to converge when P_g and P_r are disjoint.

Arjovsky et al. (2017) provided a new perspective on understanding the gradient vanishing problem. They argued that gradient vanishing stems from the ill-behaving of traditional metrics, i.e., the distance between P_g and P_r remains constant when they are disjoint. Wasserstein distance is thus proposed as an alternative metric, which can properly measure the distance between two distributions no matter they are disjoint or not. However, as we will show in the next sections, Wasserstein distance still suffers from the same problem on $\nabla_x f^*(x)$ as traditional GANs.

In summary, gradient vanishing is about the scalar term $\nabla_{f(x)}\psi(f(x))$ in $\nabla_x J_G(x)$ or the overall scale of $\nabla_x J_G(x)$, and in this paper we investigate its direction $\nabla_x f^*(x)$, where the problem is so fundamental and challenging that even the Wasserstein distance, which can properly measure the distance for disjoint distributions, also suffers from the same issue.

[2]Here and later, "two distributions are disjoint" means that their supports are disjoint.

2.3 WASSERSTEIN DISTANCE SUFFERS FROM THE SAME PROBLEM

The (1st-)Wasserstein distance is a distance function defined between two probability distributions:

$$W_1(P_g, P_r) = \inf_{\pi \in \Pi(P_g, P_r)} \mathbb{E}_{(x,y)\sim\pi(P_g,P_r)} \left[d(x,y)\right], \tag{7}$$

where $\Pi(P_r, P_g)$ denotes the collection of all probability measures with P_r and P_g being the marginal distributions of the first and second factors respectively. Since solving it in the primal form (Eq. (7)) is burdensome, Wasserstein distance is usually solved in its dual form (Villani, 2008):

$$\begin{aligned} W_1(P_g, P_r) &= \sup_f \mathbb{E}_{x\sim P_g}\left[f(x)\right] - \mathbb{E}_{x\sim P_r}\left[f(x)\right], \\ \textit{s.t. } & f(x) - f(y) \leq d(x,y),\ \forall x \sim P_g, \forall y \sim P_r. \end{aligned} \tag{8}$$

We leave the detailed discussion on the relationship between Lipschitz-continuity and Wasserstein distance in Section 4.4. Eq. (8) is the exact form of Wasserstein distance in duality, in which we remove the unnecessarily stronger Lipschitz constraint and keep a pure form of Wasserstein distance. This is to demonstrate that a well-defined distance metric may also suffer from the same problem in $\nabla_x f^*(x)$ and does not necessarily ensure the convergence of GANs.

We now study the optimal discriminative function $f^*(x)$ of Wasserstein distance. Since there is generally no closed-form solution for $f^*(x)$ in Wasserstein distance, we use an illustrative example for demonstration here, but the conclusion is general. Let $Z \sim U[0,1]$ be a uniform variable on interval $[0,1]$, P_g be the distribution of $(1, Z) \in \mathbb{R}^2$, and P_r be the distribution of $(0, Z) \in \mathbb{R}^2$, as shown in Figure 1. According to Eq. (8), one of the optimal f^* is as follows

$$f^*(x) = \begin{cases} 0 & \forall x \sim P_g, \\ 1 & \forall x \sim P_r. \end{cases} \tag{9}$$

Though having the constraint "$f(x) - f(y) \leq d(x,y), \forall x \sim P_g, \forall y \sim P_r$", Wasserstein distance also only defines the value of $f^*(x)$ on the supports of P_g and P_r, and the values of $f^*(x)$ on P_g contain no useful information about the location of P_r. Therefore, if P_g and P_r are disjoint, $\nabla_x f^*(x)$ hardly provides useful information to the generator about "how to change P_g into P_r" and the generator is not guaranteed to converge to the case $P_g = P_r$. It is worth noticing that in Wasserstein distance, the value of $f^*(x)$ on the region other than the supports of P_g and P_d is still undefined (see Figure 1b).

2.4 "$P_g = P_r$ IS THE OPTIMUM" NOT NECESSARILY GUARANTEES THE CONVERGENCE

The objectives of GANs are usually defined as (or proved equivalent to) minimizing a distance metric between P_g and P_r, which implies that "$P_g = P_r$ is the unique global optimum," and is in accordance with the final goal of the generative model, i.e., estimating the distribution of real samples. In this section, however, we emphasize that "$P_g = P_r$ is the optimum" is not enough for guaranteeing the convergence of GANs.

Given an objective is convex with respect to P_g and holds the property that "$P_g = P_r$ is the unique optimum," the convergence of GANs is guaranteed if only they are directly optimizing P_g, i.e., "dragging" P_g to P_r. However, what they actually do is using $\nabla_x f^*(x)$ as the guidance to optimize the generated samples. As shown in previous sections, when P_g and P_r are disjoint, $\nabla_x f^*(x)$, i.e., the direction that the generator follows for updating the generated samples, tells nothing about how to pull P_g to P_r. In this way, the convergence of GANs are not necessarily guaranteed. $\nabla_x f^*(x)$ indeed indicates the direction of decreasing the objective in terms of the current $f^*(x)$, but updating x to make the value of $f^*(x)$ increase / decrease does not necessarily imply that P_g is getting closer to P_r. Recall that in the failure case of Wasserstein distance in the above section, the values of $f^*(x)$ on P_g is 0, while the values of $f^*(x)$ around P_g is undefined.

In conclusion, "$P_g = P_r$ is the optimum" does not guarantee the convergence and sample updating according to $\nabla_x f^*(x)$ does not necessarily decrease the distance between P_g and P_r. Therefore, if we use $\nabla_x f^*(x)$ for updating the generator, it is necessary to make $\nabla_x f^*(x)$ aware of how to pull P_g to P_r. Alternative strategies actually exist, for example, Seguy et al. (2017) use the optimal transport plan between P_g and P_r to update the generator; however, their results tend to be blurry. In the next section, we will introduce the Lipschitz constraint as a general solution for making $\nabla_x f^*(x)$ well-behaving and guaranteeing the convergence of $\nabla_x f^*(x)$-based GANs.

3 A GENERAL SOLUTION: LIPSCHITZ CONDITION

Lipschitz-continuity constraint becomes popular in GANs recently. The first attempt of introducing Lipschitz constraint in GANs is the Wasserstein GAN (Arjovsky et al., 2017), and later Lipschitz constraint is extended to other objectives such as (Kodali et al., 2017; Fedus et al., 2017; Miyato et al., 2018), achieving great success. In this section, we explain the significance of Lipschitz constraint when introduced into the objective of the discriminator. In a nutshell, under a board family of GAN objectives, Lipschitz constraint is able to connect P_g and P_r through $f^*(x)$ such that when P_r and P_g are disjoint, $\nabla_x f^*(x)$ for each generated sample $x \sim P_g$ will point towards some real sample $y \sim P_r$, which guarantees the trend that "P_g is getting closer to P_r at every step".

3.1 THE MAIN RESULT

A function $f : X \to Y$ is k-Lipschitz continuous if it satisfies the following property:

$$d_Y(f(x), f(y)) \leq k \cdot d_X(x, y), \forall\, x, y \in X, \tag{10}$$

where d_X and d_Y are distances metrics in domains X and Y, respectively. The smallest constant k is called the Lipschitz constant. In this paper (and most GAN papers), d_X and d_Y are defined as Euclidean distance.[3] We let $\|y-x\|$ denote Euclidean distance.

As proved by Gulrajani et al. (2017), when the Lipschitz condition is combined with Wasserstein distance, we have the following property if $f^*(x)$ is differentiable, then

$$\Pr\left(\nabla_x f^*(x_t) = \frac{y-x}{\|y-x\|}\right) = 1, \text{ for } (x, y) \sim \pi^*, \tag{11}$$

where $x_t = tx + (1-t)y$, $0 \leq t \leq 1$, and π^* is the optimal π in Eq. (7). The meaning of this proposition is two-fold: (i) for each $x \sim P_g$, there exists a $y \sim P_r$ such that $\nabla_x f^*(x_t) = \frac{y-x}{\|y-x\|}$, for all linear interpolations x_t between x and y; (ii) these (x, y) pairs match the optimal coupling π^*.

Next we introduce our theorem on the Lipschitz condition. It turns out when combining the Lipschitz condition with generalized objectives, Property-(i) still holds and Property-(ii) is naturally dismissed as it is now not restricted to Wasserstein distance.

Theorem 1. Let $\bar{P}_g$ and $\bar{P}_r$ denote the supports of P_g and P_r, respectively. Assume $f^* = \arg\min_f [J_D + \lambda \cdot k(f)^2]$, where $k(f)$ is the Lipschitz constant of f. If $\phi(x)$ and $\varphi(x)$ in J_D satisfy

$$\begin{cases} \phi'(x) > 0, \phi''(x) \geq 0, \\ \varphi'(x) < 0, \varphi''(x) \geq 0, \\ \exists\, a,\ \phi'(a) + \varphi'(a) = 0, \end{cases} \tag{12}$$

then we have that

(a) $\forall x \in \bar{P}_g \cup \bar{P}_r$, $\exists y_{\neq x} \in \bar{P}_g \cup \bar{P}_r$ such that $|f^*(y) - f^*(x)| = k(f^*) \cdot \|x-y\|$ or $\nabla_{f^*(x)} J_D(x) = 0$;

(b) $\forall x \in \bar{P}_g \cup \bar{P}_r - \bar{P}_g \cap \bar{P}_r$, $\exists y_{\neq x} \in \bar{P}_g \cup \bar{P}_r$ such that $|f^*(y) - f^*(x)| = k(f^*) \cdot \|x-y\|$;

(c) if $\bar{P}_g = \bar{P}_r$ and $P_g \neq P_r$, then $\exists x$, $\exists y_{\neq x}$ such that $|f^*(y) - f^*(x)| = k(f^*) \cdot \|x-y\|$;

(d) the only Nash Equilibrium of $J_D + \lambda \cdot k(f)^2$ is reached when $P_g = P_r$, where $k(f) = 0$.

The above theorem states that when the Lipschitz condition is combined with an objective that satisfies Eq. (12), then: (a) for the optimal discriminative function $f^*(x)$ at any point $x \in \bar{P}_g \cup \bar{P}_r$, it either is bounded by the Lipschitz constant or $J_D(x)$ holds a zero-gradient with respect to $f^*(x)$; (b) for any point that only appears in $\bar{P}_g$ or $\bar{P}_r$, there must exist a point that bounds this point in terms of $|f^*(y) - f^*(x)| = k(f^*) \cdot \|x-y\|$, because for these points, $J_D(x)$ will never get zero gradient with respect to $f^*(x)$ as we prove in the Appendix G; (c) when P_g and P_r are totally overlapped, as long as P_g still not converges to P_r, there exists at least one pair (x, y) that bounds each other; (d) the only Nash Equilibrium among P_g and $f^*(x)$ under this objective is "$P_g = P_r$ with $k(f^*) = 0$." The formal proof is in Appendix G.

[3] Actually, we argue that the distance metrics must be Euclidean distance in GANs. See Appendix D.

The Wasserstein distance, i.e., $\phi(x) = \varphi(-x) = x$ is one instance that satisfies Eq. (12); and it is a very special case, which holds $\phi''(x) = 0$ and $\varphi''(x) = 0$. Eq. (12) is actually quite general and there exists many other settings, e.g., $\phi(x) = \varphi(-x) = -\log(\sigma(-x))$, $\phi(x) = \varphi(-x) = x + \sqrt{x^2+1}$, $\phi(x) = \varphi(-x) = \exp(x)$, etc. Generally, it is feasible to set $\phi(x) = \varphi(-x)$. As such, to build a new objective, one only needs to find a function that is increasing and has non-decreasing derivative. In addition, all linear combinations of feasible (ϕ, φ) pairs also lie in the family.

It is worth noting that $k(f)$ is also optimized here and it is actually necessary for Property-(c) and Property-(d). This is the key difference when the Lipschitz condition is extended to general objectives. The underlying reason for the need of also minimizing $k(f)$ comes from the existence of the case "$\nabla_{f^*(x)} J_D(x) = 0$ for $P_g(x) \neq P_r(x)$", which does not hold when the objective is Wasserstein distance. Minimizing $k(f)$ guarantees that the only Nash Equilibrium is "$P_g = P_r$ with $k(f^*) = 0$." On the other hand, if $k(f)$ is not minimized towards zero, Wasserstein distance based GANs are not guaranteed to have zero gradient $\nabla_{f^*(x)}$ at the convergence state $P_g = P_r$. It indicates that minimizing $k(f)$ is also beneficial to the Wasserstein GAN (Arjovsky et al., 2017).

3.2 Lipschitz condition connects P_g and P_r via $f^*(x)$

From Theorem 1, we know that for any point x, as long as $J_D(x)$ does not hold a zero gradient with respect to $f^*(x)$, $f^*(x)$ must be bounded by another point y such that $|f^*(y) - f^*(x)| = k(f^*) \cdot \|x - y\|$. We here further clarify that, when there is a bounding relationship, it must involve both real sample(s) and fake sample(s). More formally, we have

Theorem 2. If $f^* = \arg\min_f [J_D + \lambda \cdot k(f)^2]$, then

- $\forall x \in \bar{P}_g$, if $\exists z_{\neq x} \in \bar{P}_g \cup \bar{P}_r$ such that $|f^*(x) - f^*(z)| = k(f^*) \cdot \|x - z\|$, then $\exists y_{\neq x} \in \bar{P}_r$ such that $f^*(y) - f^*(x) = k(f^*) \cdot \|x - y\|$,
- $\forall y \in \bar{P}_r$, if $\exists z_{\neq y} \in \bar{P}_g \cup \bar{P}_r$ such that $|f^*(z) - f^*(y)| = k(f^*) \cdot \|z - y\|$, then $\exists x_{\neq y} \in \bar{P}_g$ such that $f^*(y) - f^*(x) = k(f^*) \cdot \|x - y\|$.

The intuition behind the above theorem is that samples from the same distribution, e.g., the fake samples, will not bound each other. It is worth noticing that there might exist a chain of bounding relationships that involves a dozen of fake samples and real samples, and these points all lie in the same line and bounds each other.

Under the Lipschitz condition, the **bounded line** in the value surface of f^* is the basic **building block** that connects P_g and P_r, and each fake sample lies in one of the bounded lines. Next we will further interpret the implication of bounding relationship and show that it guarantees meaningful $\nabla_x f^*(x)$ for all involved points.

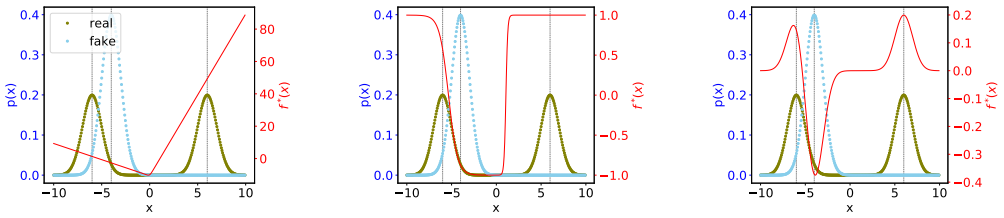
3.3 Lipschitz condition ensures the convergence of $\nabla_x f^*(x)$-based GANs

Recall that the proposition in Eq. (11) states that $\nabla_x f^*(x_t) = \frac{y-x}{\|y-x\|}$. This is actually a direct consequence of bounding relationship between x and y. We formally state it as follows:

Theorem 3. Assume $f(x)$ is differentiable and k-Lipschitz continuous. If $f(y) - f(x) = k \cdot \|x - y\|$ $\forall x \neq y$, then $\nabla_x f(x_t) = k \cdot \frac{y-x}{\|x-y\|}$, where $x_t = tx + (1-t)y$ for $0 \leq t \leq 1$.

In other words, if two points x and y bound each other in terms of $f(y) - f(x) = k \cdot \|x - y\|$, there is a straight line between x and y in the value surface of f. Any point in this line holds the maximum gradient slope k, and the direction of these gradient all point towards the $x \to y$ direction. The proof is provided in Appendix D. Combining Theorem 1 and Theorem 2, we can conclude that when P_g and P_r are disjoint, $\nabla_x f^*(x)$ for each sample $x \sim P_g$ points to a sample $y \sim P_r$, which guarantees that P_g is moving towards P_r.

In fact, Theorem 1 provides further guarantee on the convergence. Property-(b) implies that for any $x \sim P_g$ that does not lies in P_r, $\nabla_x f^*(x)$ points to some real sample $y \sim P_r$. In the fully overlapped case, according to Property-(c), unless $P_g = P_r$, there exists a pair (x, y) in bounding relationship and $\nabla_x f^*(x)$ pulls x towards y. Property-(d) guarantees that the only Nash Equilibrium is "$P_g = P_r$."



(a) Original GAN (b) Least Square GAN (c) Fisher GAN with uniform μ

Figure 2: The source of Mode Collapse. In traditional GANs, $f^*(x)$ is a function of the local densities $P_g(x)$ and $P_r(x)$. Given $f^*(x)$ is an increasing function of $P_r(x)$ and decreasing function of $P_g(x)$, when fake samples get close to a mode of the P_r, $\nabla_x f^*(x)$ move them towards the mode.

4 Extensions and Discussions

4.1 From disjoint case to overlapping case

In Section 2, we discuss the problem of $f^*(x)$ and $\nabla_x f^*(x)$ in the case where P_g and P_r are disjoint. In this section, we extend our discussion to the overlapping case. In the disjoint case, we argue that "$f^*(x)$ on P_g does not reflect any information about the location of other points in P_r" will lead to an unfeasible $\nabla_x f^*(x)$ and thus non-convergence. In the overlapping and continuous case, things are actually different, $f^*(x)$ around each point is also defined, and its gradient $\nabla_x f^*(x)$ now reflects the local variation of $f^*(x)$.

For most traditional GANs, $f^*(x)$ mainly reflects the local information about the density $P_g(x)$ and $P_r(x)$. However, it is worth noting that $f^*(x)$ is usually an increasing function with respect to $P_r(x)$ while a decreasing function with respect to $P_g(x)$. For instance, $f^*(x)$ in the original GAN is $\log P_r(x)/P_g(x)$. Optimizing the generator according $\nabla_x f^*(x)$ will move sample x towards the direction of increasing $f^*(x)$. Because $f^*(x)$ positively correlates with $P_r(x)$ and negatively correlated with $P_g(x)$, it in sense means x is becoming more real. However, such a local greedy strategy turns out to be a fundamental cause of mode collapse.

4.2 The cause of mode collapse: the locality of $f^*(x)$ and ∇

Mode collapse is a notorious problem in GANs' training, which refers to the phenomenon that the generator only learns to produce part of P_r. Many literatures try to study the source of mode collapse (Che et al., 2016; Metz et al., 2016; Kodali et al., 2017; Arora et al., 2017) and measure the degree of mode collapse (Odena et al., 2016; Arora & Zhang, 2017).

The most recognized cause of mode collapse is that, if the generator is much stronger than the discriminator, it may learn to only produce the sample(s) in the local or global maximum of $f(x)$ for the current discriminator. This argument is true for most of GAN models. However, from our perspective on $f^*(x)$ and its gradient, there actually exists a much more fundamental cause of mode collapse, i.e., the locality of $f^*(x)$ in traditional GANs and the locality of gradient operator ∇.

In traditional GANs, $f^*(x)$ is a function of local densities $P_g(x)$ and $P_r(x)$, which is local, and the gradient operator ∇ is also a local operator. As the result, $\nabla_x f^*(x)$ only reflects its local variations and cannot capture the statistic of P_r and P_g that is far from itself. If $f^*(x)$ in the surrounding area of x is well-defined, $\nabla_x f^*(x)$ will move x towards the **nearby** location where the value of $f^*(x)$ is higher. It does not take the global status into account.

The typical result is that when fake samples get close to a mode of the P_r, they move towards the mode and get stuck there (due to the locality). Assume P_r consists of two Gaussian distributions (A and B) that are distant from each other, while the current P_g is uniformly distributed over its support and close to real Gaussian A. In this case, $\nabla_x f(x)$ of all fake samples will point towards the center of Gaussian A. If P_g is a Gaussian with the same standard deviation as Gaussian A, $\nabla_x f(x)$ in original GAN and Least-Square GAN shows almost identical behaviors, which is illustrated in Figure 2. In Fisher GAN, if $\mu(x)$ is uniform, the case is even worse: a large amount of points that are relatively far from Gaussian A will move away from A (but the direction is not necessarily towards B, though in our 1-D case it is).

This observation again supports our argument: "$P_g = P_r$ is optimum" is not enough and the validity of $\nabla_x f^*(x)$ is also necessary (even if P_g and P_r is continuous and overlapped). As far as we know, this is the first work that provides a clear explanation on this cause of mode collapse.

4.3 Explanation on the empirical success of traditional GANs

Though traditional GANs does not have any guarantee on its convergence, it has already achieved its great success. The reason is that having no guarantee does not mean it cannot converge. It turns out extensive parameter-tuning actually increases the probability of the convergence.

As shown in Appendix A, hyper-parameters are important in influencing the value surface of $f^*(x)$. Some typical settings (e.g., simplified neural network architecture, relu or leaky relu activation, relatively high learning rate, Adam optimizer, etc.) tend to form a relatively smooth value surface (e.g., monotonically increasing from P_g to P_r), making $\nabla_x f^*(x)$ much more meaningful. That is, one can find these settings, where $\nabla_x f^*(x)$ or $\nabla_x f(x)$ is more favourable, to enable traditional GAN to work. In opposite, we have tried highly-nonlinear activation such as swish (Ramachandran et al., 2018) in the discriminator. It turns out traditional GANs is very likely to fail. In contrast, our proposed Lipschitz constraint based GAN is compatible with highly-nonlinear activation. Another important empirical technique is to delicately balance the generator and the discriminator or limit the capacity of the discriminator. This is to avoid the fatal optimal $f^*(x)$. All these could possibly make traditional GANs work. However, the consequence is that these GANs are very sensitive to hyper-parameters and hard to use.

4.4 Lipschitz condition is stronger than the one in Wasserstein distance

Most literature directly writes the dual form of Wasserstein distance with the Lipschitz condition. However, it is worth noticing that the Lipschitz condition is actually stronger than the necessary one in the dual form of Wasserstein distance. Recall that the necessary constraint in dual form of Wasserstein distance (Section 2.3) is

$$f(x) - f(y) \leq d(x, y),\ \forall x \sim P_g, \forall y \sim P_r. \tag{13}$$

However, the 1-Lipschitz constraint is

$$f(x) - f(y) \leq d(x, y), \forall x, \forall y. \tag{14}$$

The key difference is that the necessary constraint (Eq. 13) restricts the range of x and y, but Lipschitz constraint (Eq. 14) does not have the restriction on the range and thus is the sufficient condition of the former one.

If the supports of P_g and P_r are the entire space, Eq. (13) and Eq. (14) are actually identical, and Wasserstein distance should also work. However, P_g and P_r are usually disjoint in GANs. Therefore, introducing a stronger Lipschitz constraint is necessary to ensure the validity of Wasserstein distance in $\nabla_x f^*(x)$-based updating.

4.5 Contribution clarification

This work is substantially different from Wasserstein GAN (Arjovsky et al., 2017). The main argument in (Arjovsky et al., 2017) is "we should switch to Wasserstein distance", which does not hold in general, as we have argued. Though the final solution in Wasserstein GAN is sound as we prove.

We have shown that Lipschitz constraint is able to ensure the convergence of GANs in a family of GAN objectives, which is not restricted to Wasserstein distance. For example, Lipschitz constraint is also introduced to original GAN in (Miyato et al., 2018) and (Kodali et al., 2017) and shows improvements on the quality of generated samples. As a matter of fact, the original GAN objective $\phi(x) = \varphi(-x) = -\log(\sigma(-x))$ is another instance in our proposed family and thus our analysis explains why and how it works.

It is also worth noticing that J_D in our formulation is not derived from any well-established distance metric; it is derived based on Lipschitz constraint. As we have shown that a well-established distance or divergence does not necessarily ensure the convergence, we believe our trial could shed light on the new direction of GANs.

Last but no least, though we do not discuss the generator's objective, our analysis indicates that the **minimax** in terms of ψ in Eq. (1) is not essential, because it only influences the scale of the gradient. Nevertheless, the function ψ does influence the updating of the generator, and we leave the detailed investigation as future work. Another example that has a theoretically meaningful $\nabla_x f^*(x)$ is Coulomb GAN (Unterthiner et al., 2017), which is also derived neither from minimax game nor from well-defined distance metric.

4.6 Related work

Fedus et al. (2017) also argued that divergence is not the primary guide of the training of GANs and pointed out that the gradient does not necessarily related to the divergence. However, they tended to believe that original GAN with non-saturating generator objective can somehow work. As we have proved before, given the optimal f^*, the original GAN has no guarantee on its convergence. And we argue that practical work scenarios benefit from parameter-tuning.

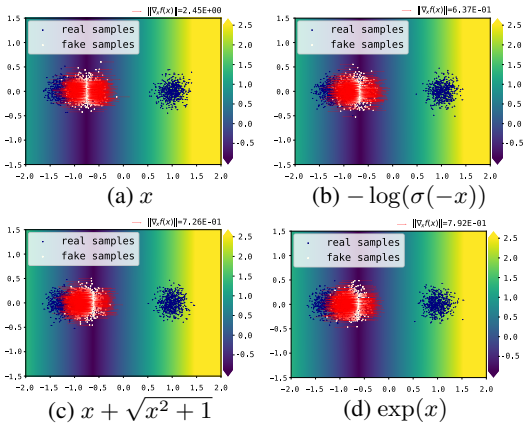
Some work study the suboptimal $f(x)$ (Mescheder et al., 2017; 2018; Arora et al., 2017), which is another important direction for understanding GANs theoretically. While the behaviors of suboptimal can be slightly different, we think the optimal $f^*(x)$ should well-behave in the first place.

Researchers also found that applying Lipschitz constraint to the generator also benefits the quality of generated samples (Zhang et al., 2018; Odena et al., 2018). In addition, researchers also investigated implementation of Lipschitz constraint in GANs (Gulrajani et al., 2017; Petzka et al., 2017; Miyato et al., 2018). However, this branch of related work is out of scope of the discussion in this paper.

5 Experiments

In this section, we present the experiment results on our proposed objectives for GANs. The anonymous code is provided at `http://bit.ly/2Kvbkje`.

5.1 Verifying the objective family and its gradient $\nabla_x f^*(x)$



(a) x (b) $-\log(\sigma(-x))$ (c) $x + \sqrt{x^2+1}$ (d) $\exp(x)$

Figure 3: Verifying the objective family

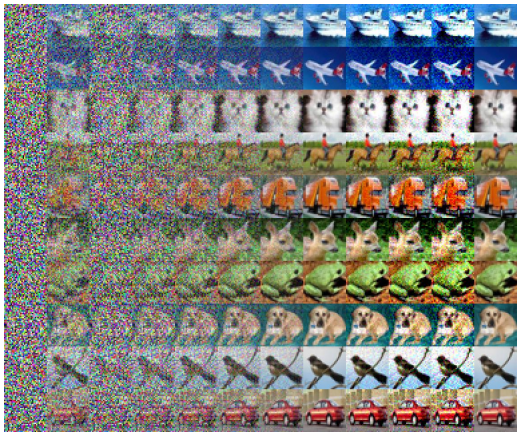
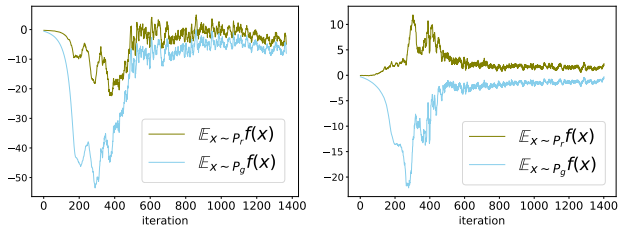


Figure 4: $\nabla_x f^*(x)$ gradation with CIFAR-10

We verify a set of ϕ and φ satisfying Eq. (12): (a) $\phi(x) = \varphi(-x) = x$; (b) $\phi(x) = \varphi(-x) = -\log(\sigma(-x))$; (c) $\phi(x) = \varphi(-x) = x + \sqrt{x^2+1}$; (d) $\phi(x) = \varphi(-x) = \exp(x)$. As shown in Figure 3, the gradient of each generated sample is towards a real sample.

We further verify $\nabla_x f^*(x)$ with the real-world data, using **ten CIFAR-10 images** as P_r and **ten noise images** as P_g to make the solving of $f^*(x)$ feasible. The result is shown in Figure 4, where The leftmost in each row are the $x \sim P_g$ and the second are their gradient $\nabla_x f(x)$. The interior are $x + \epsilon \cdot \nabla_x f(x)$ with increasing ϵ, which will pass through a real sample, and the rightmost are the nearest $y \sim P_r$. This result visually demonstrates that the gradient of a generated sample is towards the direction of one real sample. Note that the final results of this experiment keep almost identical when varying the loss metric $\phi(x)$ and $\varphi(x)$ in the family.



(a) x (b) $-\log(\sigma(-x))$

Figure 5: $f^*(x)$ in new objective is more stable.

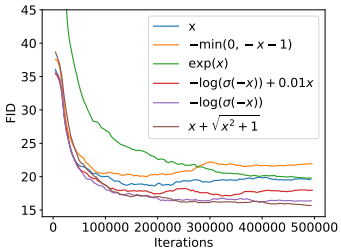


Figure 6: Training curves on CIFAR-10.

5.2 Stabilizing $f^*(x)$ with new Objectives

Wasserstein distance is a special case in our proposed family of objectives where $\phi''(x) = \varphi''(x) = 0$. As a result, $f^*(x)$ under the Wasserstein distance objective where $\phi(x) = \varphi(-x) = x$ has a free offset, which means given a $f^*(x)$, $f^*(x) + b$ with any $b \in \mathbb{R}$ is also an optimal. In practice, this behaves as an oscillatory $f(x)$ during training. Any other instance in our new proposed objectives does not have this problem. We illustrate this practical difference in Figure 5.

5.3 Benchmark on unsupervised image generation tasks

Table 2: Quantitative comparisons on unsupervised image generation tasks.

Objective	CIFAR-10		Tiny ImageNet		Oxford 102	
	FID	IS	FID	IS	FID*	IS*
$-\min(0, -x-1)$	21.58 ± 0.21	7.43 ± 0.04	16.22 ± 0.33	$\mathbf{8.58 \pm 0.08}$	9.72 ± 0.51	$\mathbf{21.91 \pm 0.18}$
x	19.64 ± 0.23	7.66 ± 0.03	18.81 ± 0.58	8.20 ± 0.05	9.74 ± 0.63	21.66 ± 0.22
$-\log(\sigma(-x))$	$\mathbf{16.36 \pm 0.09}$	$\mathbf{8.49 \pm 0.11}$	$\mathbf{15.94 \pm 0.33}$	8.42 ± 0.04	$\mathbf{9.40 \pm 0.49}$	21.82 ± 0.11
$x + \sqrt{x^2+1}$	$\mathbf{15.76 \pm 0.13}$	$\mathbf{8.04 \pm 0.04}$	16.83 ± 0.41	8.35 ± 0.09	$\mathbf{9.16 \pm 0.52}$	$\mathbf{21.96 \pm 0.19}$
$\exp(x)$	19.82 ± 0.13	7.79 ± 0.03	20.45 ± 0.15	8.06 ± 0.05	9.90 ± 0.72	21.91 ± 0.22
$-\log(\sigma(-x)) + 0.01x$	18.32 ± 0.15	7.75 ± 0.04	$\mathbf{16.09 \pm 0.23}$	$\mathbf{8.47 \pm 0.10}$	9.50 ± 0.39	21.91 ± 0.20

Finally, we fix $\psi(x) = -x$ in the generator's objective and compare various objectives on unsupervised image generation tasks. The results of Inception Score (Salimans et al., 2016) and Frechet Inception Distance (Heusel et al., 2017b) are presented in Table 2. We also include the hinge loss $\phi(x) = \varphi(-x) = -\min(0, -x-1)$ which used in (Miyato et al., 2018).

The gradient of $\exp(x)$ varies significantly and we find it requires a small learning rate to avoid explosion. The objectives $x + \sqrt{x^2+1}$, $-\log(\sigma(-x))$ and $-\log(\sigma(-x)) + 0.1x$ achieve the best performances. This is probably because they have bounded gradient and reduce the gradient of well-identified points towards zero, which enables the discriminator to pay more attention to these ill-identified. Hinge loss $-\min(0, -x-1)$ does not lie in our proposed objective family and turns out to be unstable and performs unsatisfactory in same cases. We also plot the training curve in terms of FID in Figure 6.

Due to page limitation, we leave the details, visual results and more experiments in the Appendix.

6 Conclusion

In this paper we have shown that the fundamental cause of failure in training of GANs stems from the unreliable $\nabla_x f^*(x)$. Specifically, when P_g and P_r are disjoint, $\nabla_x f^*(x)$ for fake sample $x \sim P_g$ tells nothing about P_r, making it impossible for P_g to converge to P_r. We have further demonstrated that even Wasserstein distance, which can properly measure the distance for two disjoint distributions, also suffers from the same problem when P_g and P_r are disjoint. This implies that "what distance metric should be used" does not touch the key of non-convergence of GANs. We have highlighted in the paper that a well-defined distance metric, or more generally, "$P_g = P_r$ is the optimum", is not enough for guaranteeing the convergence of GANs. Therefore, if we update the generator based on $\nabla_x f^*(x)$, we need to make much attention on the design of $f^*(x)$. Furthermore, to address the aforementioned problem, we have proposed the Lipschitz-continuity condition as a general solution to make $\nabla_x f^*(x)$ reliable and ensure the convergence of GANs, which works well in a large family of GAN objectives. In addition, we have shown that in the overlapping case, $\nabla_x f^*(x)$ is also problematic which turns out to be an intrinsic cause of mode collapse in traditional GANs.

REFERENCES

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.

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

Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. *arXiv preprint arXiv:1706.08224*, 2017.

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

Andrew Brock, Theodore Lim, JM Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.

Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016.

William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: Gans do not need to decrease adivergence at every step. *arXiv preprint arXiv:1710.08446*, 2017.

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

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

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

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 *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017a.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017b.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

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

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. In *Advances in Neural Information Processing Systems*, pp. 1825–1835, 2017.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, pp. 3478–3487, 2018.

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

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Youssef Mroueh and Tom Sercu. Fisher gan. In *Advances in Neural Information Processing Systems*, pp. 2510–2520, 2017.

Youssef Mroueh, Chun-Liang Li, Tom Sercu, Anant Raj, and Yu Cheng. Sobolev gan. *arXiv preprint arXiv:1711.04894*, 2017.

Anh Nguyen, Jason Yosinski, Yoshua Bengio, Alexey Dosovitskiy, and Jeff Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. *arXiv preprint arXiv:1612.00005*, 2016.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016.

Augustus Odena, Jacob Buckman, Catherine Olsson, Tom B Brown, Christopher Olah, Colin Raffel, and Ian Goodfellow. Is generator conditioning causally related to gan performance? *arXiv preprint arXiv:1802.08768*, 2018.

Henning Petzka, Asja Fischer, and Denis Lukovnicov. On the regularization of wasserstein gans. *arXiv preprint arXiv:1709.08894*, 2017.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. 2018.

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

Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. *arXiv preprint arXiv:1711.02283*, 2017.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, Martin Heusel, Hubert Ramsauer, and Sepp Hochreiter. Coulomb gans: Provably optimal nash equilibria via potential fields. *arXiv preprint arXiv:1708.08819*, 2017.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

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

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

A EXPERIMENTS: THE INFLUENCE OF HYPER-PARAMETERS

The value surface of traditional GANs is highly depends on the network and training hyper-parameters. It also suggests why traditional GANs are highly unstable and sensitive to hyper-parameters.

We here plot the Least-Square GAN value surface with various hyper-parameter settings, to give directly impression on how these parameters influence GANs training. Not very strictly, but our empirical code is: (i) a low-capacity network tends to learn a simple surface; (ii) SGD tends to learn a more complex surface than ADAM; (iii) large learning rate tends to learns a simpler surface than small learning rate; (iv) highly nonlinear activation function tends to result into more complex value surface.

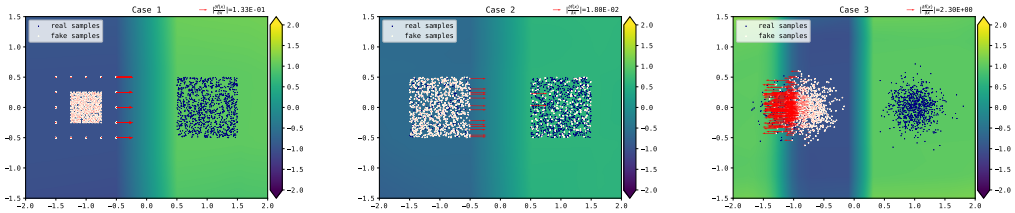


Figure 7: ADAM optimizer with lr=1e-2, beta1=0.0, beta2=0.9. MLP with RELU activations, #hidden units=1024, #layers=1.

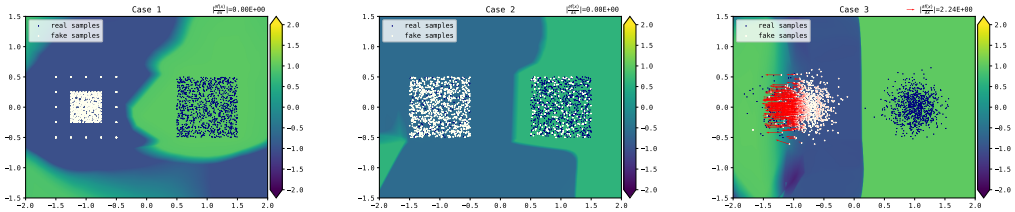


Figure 8: ADAM optimizer with lr=1e-2, beta1=0.0, beta2=0.9. MLP with RELU activations, #hidden units=1024, #layers=4.

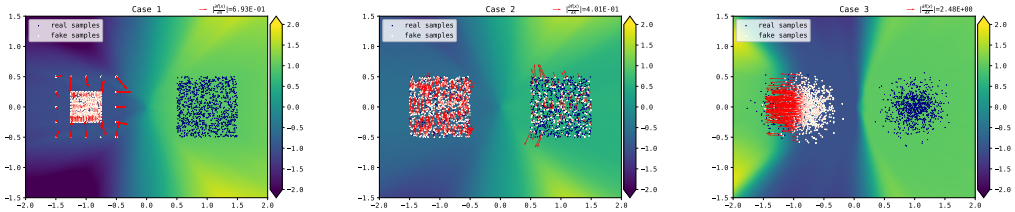


Figure 9: ADAM optimizer with lr=1e-5, beta1=0.0, beta2=0.9. MLP with RELU activations, #hidden units=1024, #layers=4.

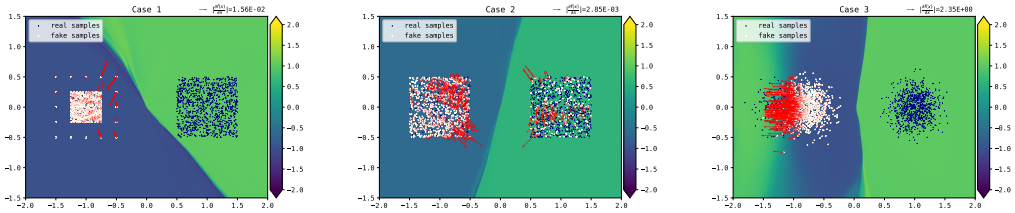


Figure 10: SGD optimizer with lr=1e-3. MLP with SELU activations, #hidden units=1024, #layers=64.

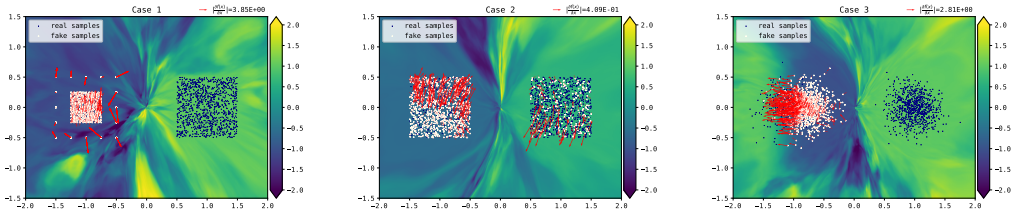


Figure 11: SGD optimizer with lr=1e-4. MLP with SELU activations, #hidden units=128, #layers=64.

B VARIOUS $\phi(x)$ AND $\varphi(x)$ THAT SATISFIES EQUATION 12

For Lipschitz constraint based GANs, $\phi(x)$ and $\varphi(x)$ are required to satisfy Equation 12. Eq. (12) is actually quite general and there exists many other instances, e.g., $\phi(x) = \varphi(-x) = x$, $\phi(x) = \varphi(-x) = -\log(\sigma(-x))$, $\phi(x) = \varphi(-x) = x + \sqrt{x^2+1}$, $\phi(x) = \varphi(-x) = \exp(x)$, etc. We plot these instances of $\phi(x)$ and $\varphi(x)$ in Figure 12.

Generally, it is feasible to set $\phi(x) = \varphi(-x)$. Note that rescaling and offsetting along the axises are trivial operation to found more $\phi(x)$ and $\varphi(x)$ within a function classes, and linear combination of two or more $\phi(x)$ or $\varphi(x)$ from different function classes also keep satisfying Equation 12.

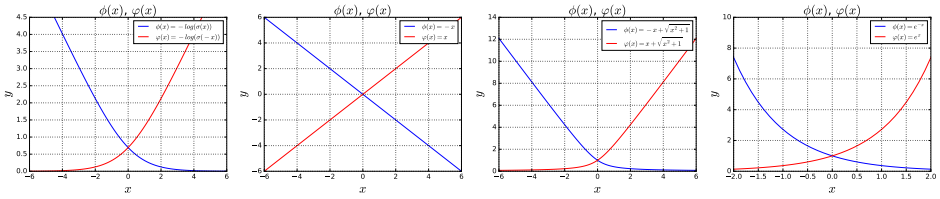
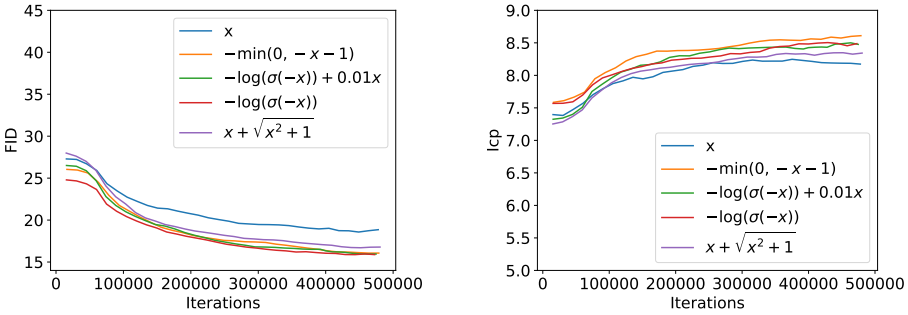


Figure 12: Various $\phi(x)$ and $\varphi(x)$ that satisfies Equation 12

C GENERATED IMAGES AND TRAINING CURVES

Extra training curves on Tiny ImageNet is proved in Figure 13. And comparisons on the visual results among different objectives are aslo provided in Figure 14, Figure 15 and Figure 16.



(a) FID training curve

(b) Inception Score training curve

Figure 13: FID and Icp(Inception Score) training curves towards different objective functions on Tiny Imagenet Dataset.

(a) $-\log(\sigma(-x))$

(b) $-\log(\sigma(-x)) + 0.01x$

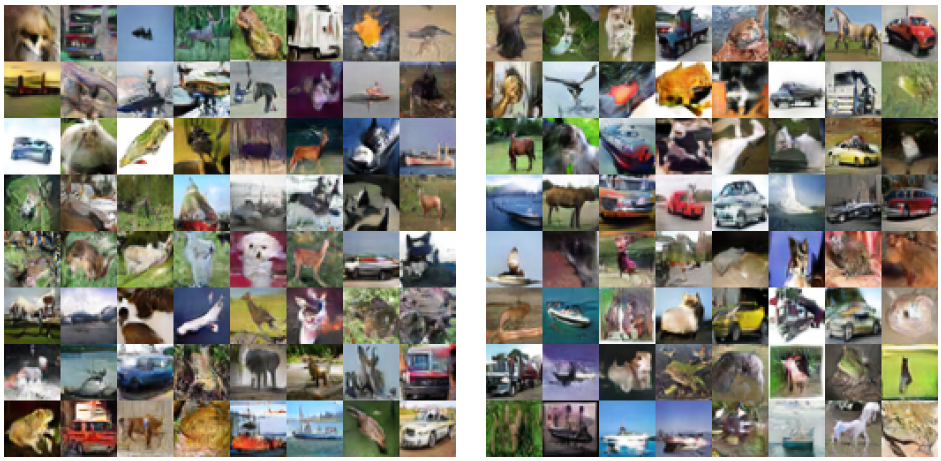
(c) x

(d) $x + \sqrt{x^2 + 1}$

(e) $\exp(x)$

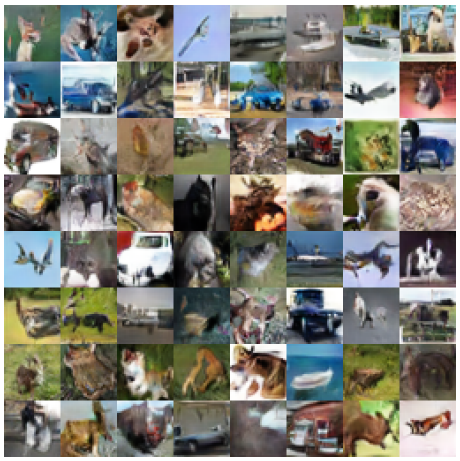
(f) $-\min(0, -x - 1)$

Figure 14: Random Samples of Lipschitz GAN trained towards different objectives on Oxford 102.

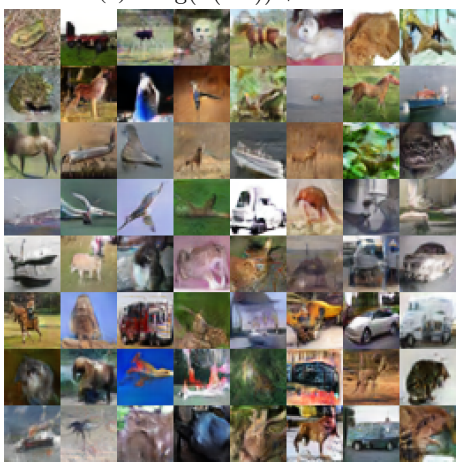


(a) $-\log(\sigma(-x))$

(b) $-\log(\sigma(-x)) + 0.01x$

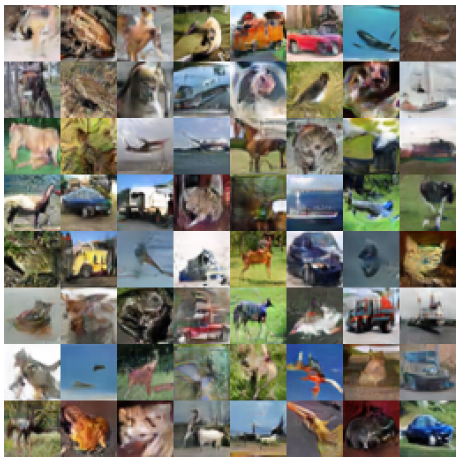


(c) x



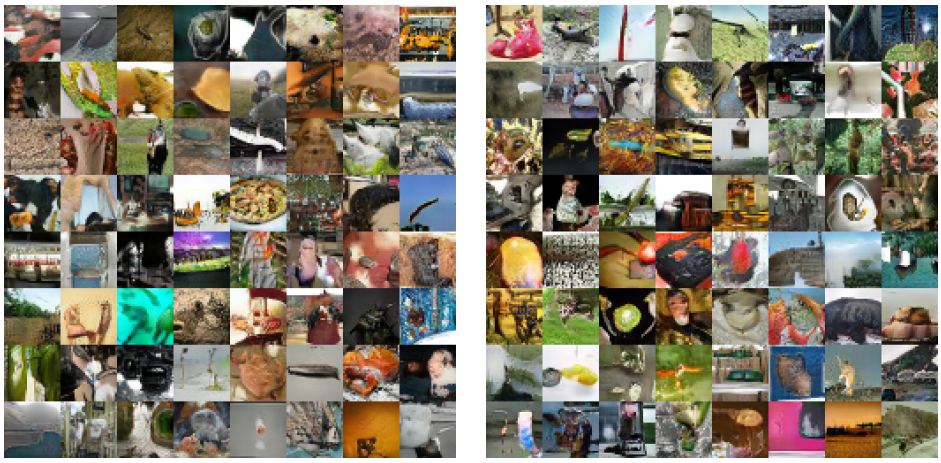
(d) $x + \sqrt{x^2 + 1}$

(e) $\exp(x)$



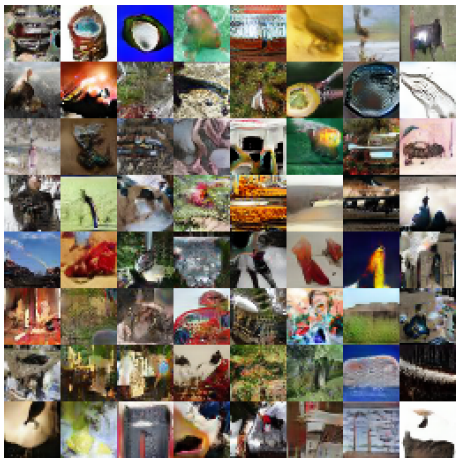
(f) $-\min(0, -x - 1)$

Figure 15: Random Samples of Lipschitz GAN trained towards different objectives on Cifar-10.

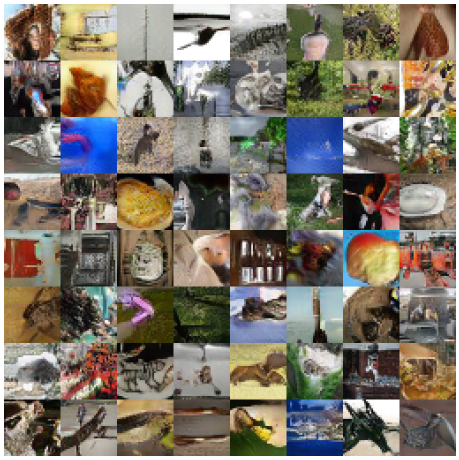


(a) $-\log(\sigma(-x))$

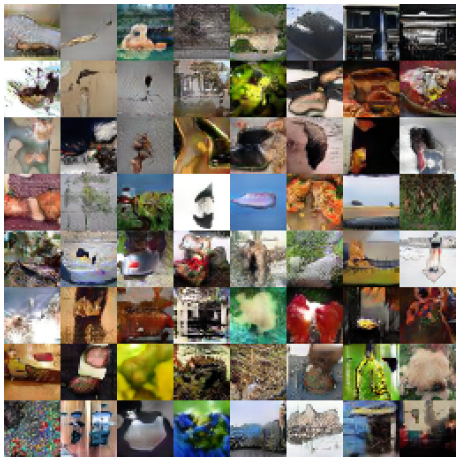
(b) $-\log(\sigma(-x)) + 0.01x$



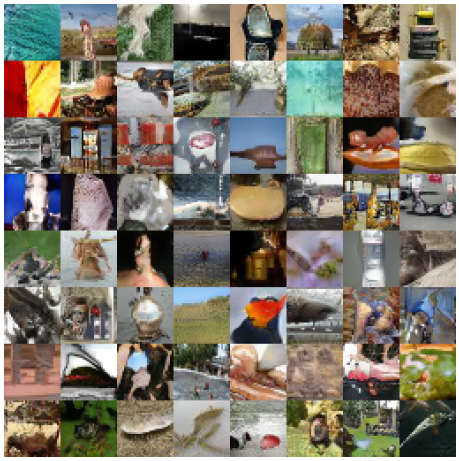
(c) x



(d) $x + \sqrt{x^2 + 1}$



(e) $\exp(x)$



(f) $-\min(0, -x - 1)$

Figure 16: Random Samples of Lipschitz GAN trained towards different objectives on Tiny Imagenet.

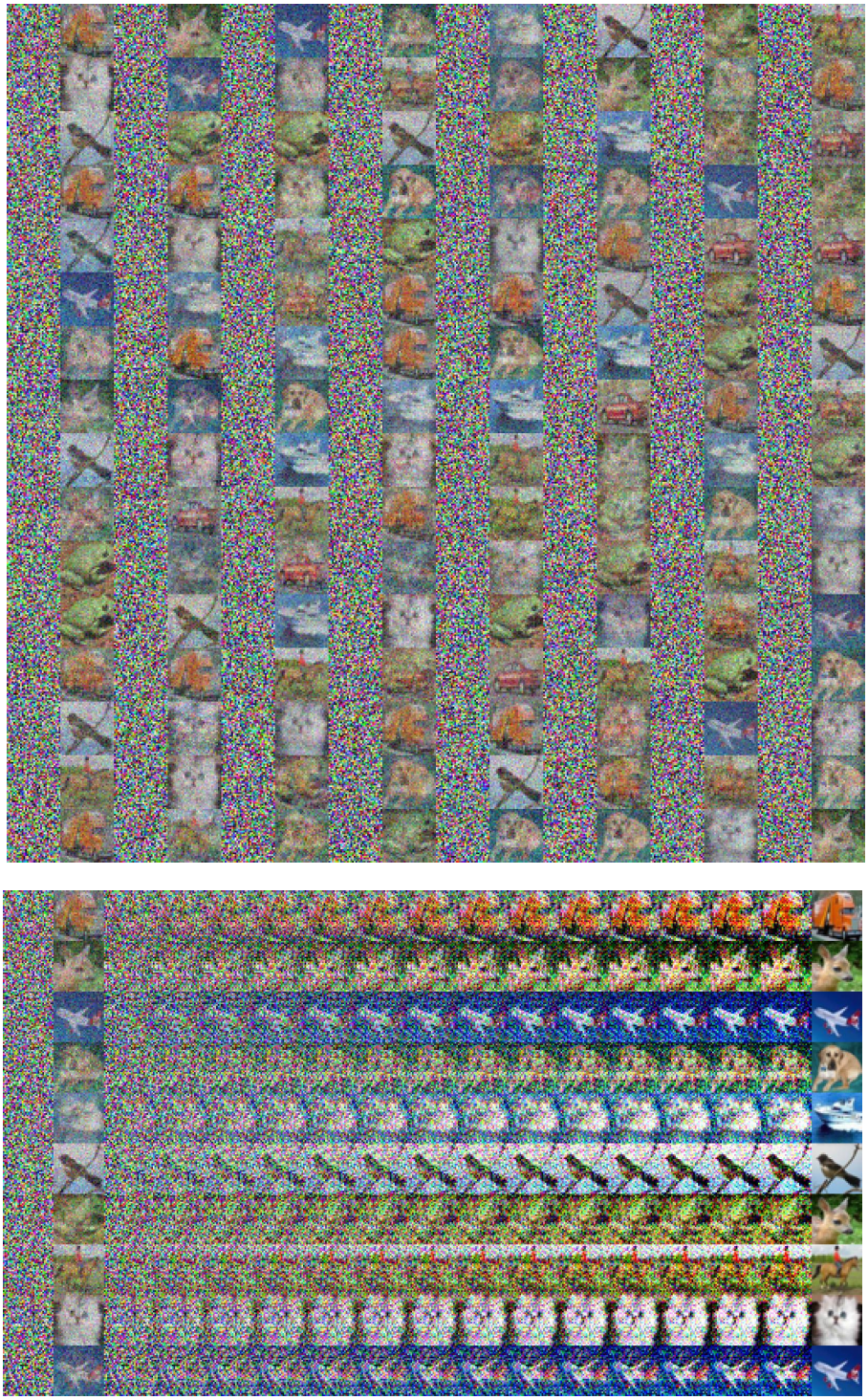


Figure 17: The gradient of Lipschitz constraint based GANs with real world data, where P_r consists of ten images and P_g is Gaussian noise. Up: Each odd column are $x \sim p_{\mathrm{G}}$ and the nearby column are their gradient $\frac{\partial f(x)}{\partial x}$. Down: the leftmost in each row are $x \sim p_{\mathrm{G}}$, the second are their gradients $\frac{\partial f(x)}{\partial x}$, the interior are $x + \epsilon \cdot \frac{\partial f(x)}{\partial x}$ with increasing ϵ, and the rightmost are the nearest $y \sim p_{\mathrm{data}}$.

D EUCLIDEAN DISTANCE IS ESSENTIAL

In this section, we delve deeply into the relationship between gradient properties and different norm in Lipschitz continuity. We will prove the Theorem 3, *i.e.* Lipschitz continuity with l_2-norm (Euclidean Distance) can guarantee the gradient direction of $\nabla f^*(x)$, and at the same time, demonstrate that the other norm does not have this property.

A brief review on Lipschitz continuity: a function f is k-Lipschitz over a set $\mathcal{S}$ with respect to a norm $\|.\|_p$, means that, for all $a, b \in \mathcal{S}$ there is $f(a) - f(b) \leq k\|a - b\|_p$. To start with, we give the proof of Theorem 3 in the following.

Proof.

Let (x, y) be such that $x \neq y$, and we define $x_t = x + t \cdot (y - x)$ with $t \in [0, 1]$. We claim that: if $f(x)$ is k-Lipschitz with respect to $\|.\|_p$ and $f(y) - f(x) = k\|x - y\|_p$, then $f(x_t) = f(x) + t \cdot k\|x - y\|_p$.

As we know $f(x)$ is k-Lipschitz, with the property of norms, we have

$$\begin{aligned} f(y) - f(x) &= f(y) - f(x_t) + f(x_t) - f(x) \\ &\leq f(y) - f(x_t) + k\|x_t - x\|_p \\ &= f(y) - f(x_t) + t \cdot k\|x - y\|_p \\ &\leq k\|y - x_t\|_p + t \cdot k\|x - y\|_p \\ &= k \cdot (1 - t)\|x - y\|_p + t \cdot k\|x - y\|_p = k\|x - y\|_p. \end{aligned} \tag{15}$$

Given $f(x) - f(y) = k\|x - y\|_p$, it implies all the inequalities need to be equalities. Therefore, $f(x_t) = f(x) + t \cdot k\|x - y\|_p$.

It is clear that: given $f(x)$ is k-Lipschitz with respect to $\|.\|_2$, if $f(x)$ is differentiable at x_t, then $\|\nabla f(x_t)\|_2 \leq k$. With $f(x_t) = f(x) + t \cdot k\|x - y\|_2$, the directional derivative of $f(x)$ on the direction $v = \frac{y-x}{\|y-x\|_2}$ at x_t is equal to k,

$$\begin{aligned} \frac{\partial f(x_t)}{\partial v} &= \lim_{h \to 0} \frac{f(x_t + hv) - f(x_t)}{h} = \lim_{h \to 0} \frac{f(x_t + h\frac{y-x}{\|y-x\|_2}) - f(x_t)}{h} \\ &= \lim_{h \to 0} \frac{f(x_{t+\frac{h}{\|y-x\|_2}}) - f(x_t)}{h} = \lim_{h \to 0} \frac{\frac{h}{\|y-x\|_2} \cdot k\|y - x\|_2}{h} = k. \end{aligned} \tag{16}$$

Note that $\|v\|_2 = \|\frac{y-x}{\|y-x\|_2}\|_2 = 1$, *i.e.* v is a unit vector. Now,

$$k^2 = k\frac{\partial f(x_t)}{\partial v} = k\langle v, \nabla f(x_t)\rangle = \langle kv, \nabla f(x_t)\rangle \leq \|kv\|_2\|\nabla f(x_t)\|_2 = k^2. \tag{17}$$

As the equality holds only when $\nabla f(x_t) = kv = k\frac{y-x}{\|y-x\|_2}$, we prove that $\nabla f(x_t) = k\frac{y-x}{\|y-x\|_2}$. □

Above proof utilizes the property that $\|\nabla f(x_t)\|_2 \leq k$, which is derived from that $f(x)$ is k-Lipschitz with respect to $\|.\|_2$. However, other norms do not hold this property. Specifically, according to the theory in Shalev-Shwartz et al. (2012): if a convex and differentiable function f is k-Lipschitz over $\mathcal{S}$ with respect to norm $\|.\|_p$, then the Lipschitz continuity actually implies a bound on the dual norm of gradients, *i.e.* $\|\nabla f\|_q \leq k$. Here $\|.\|_q$ is the dual norm of $\|.\|_p$, which satisfies the equation that $\frac{1}{p} + \frac{1}{q} = 1$.

As we could notice, a norm is equal to its dual norm if and only if $p = 2$. Switching to l_p-norm with $p \neq 2$, it is actually bounding the l_q-norm of the gradients. However, bounding the l_q-norm of the

gradients does not guarantee the gradient direction at fake samples point towards real samples. A counter-example is provided as follows.

Consider a function $g(x, y) = x + y$ on $\mathbb{R}^2$. $\forall\, p_1 = (x_1, y_1), p_2 = (x_2, y_2)$, there is $g(p_1) - g(p_2) = g(x_1, y_1) - g(x_2, y_2) = (x_1 - x_2) + (y_1 - y_2) \leq |x_1 - x_2| + |y_1 - y_2| = \left\|p_1 - p_2\right\|_1$, which means g is a 1-Lipschitz function with respect to l_1-norm. According to above analysis, the dual norm of ∇g is bounded, *i.e.* $\left\|\nabla g\right\|_\infty \leq 1$. Actually ∇g is equal to $(1, 1)$ at every point in $\mathbb{R}^2$ with $\left\|\nabla g\right\|_\infty = 1$. Selecting two points $A = (0, 0)$ and $B = (2, 1)$, we have $g(A) - g(B) = \left\|A - B\right\|_1$, however, $\nabla g(A) = (1, 1)$ is not pointing towards B.

E On the implementation of k-Lipschitz for GANs

Typical techniques for enforcing k-Lipschitz includes: spectral normalization Miyato et al. (2018), gradient penalty Gulrajani et al. (2017), and Lipschitz penalty Petzka et al. (2017). Before moving into the detailed discussion of these methods, we would provide several important notes in the first place.

Firstly, enforcing k-Lipschitz in the blending-region of p_{G} and p_{data} is actually sufficient. Define $B(\mu, \nu) = \{\hat{x} = x \cdot t + y \cdot (1 - t) \mid x \sim \mu \wedge y \sim \nu \wedge t \in [0, 1]\}$. It is clear that $f(x)$ is 1-Lipschitz in $B(\mu, \nu)$ implies $f(x) - f(y) \leq d(x, y), \forall x \in \mu, \forall y \in \nu$. Thus, it is a sufficient constraint for Wasserstein distance in Equation 8. In fact, $f(x)$ is k-Lipschitz in $B(p_{\mathrm{G}}, p_{\mathrm{data}})$ is also a sufficient condition for all properties described in Lipschitz constraint based GANs (Section 3).

Secondly, enforcing k-Lipschitz with regularization would provide a dynamic Lipschitz constant k.

Theorem 4. With Wasserstein GAN objective, we have $\min_{f \in \mathcal{F}_{\text{k-Lip}}} J_D(f) = k \cdot \min_{f \in \mathcal{F}_{\text{1-Lip}}} J_D(f)$.

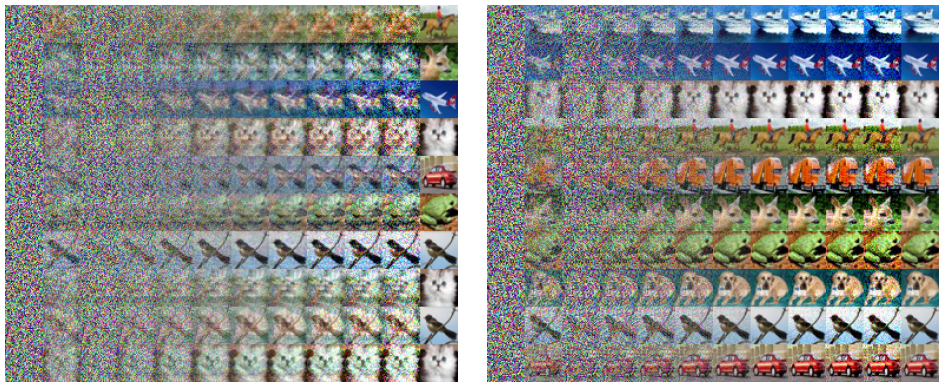
Assuming we know and can control the Lipschitz constant k of $f(x)$, by introducing a loss, saying square loss, on k respecting to a constant k_0, the total loss of the discriminator (critic) becomes $J(k) \triangleq \min_{f \in \mathcal{F}_{\text{k-Lip}}} J_D(f) + \lambda \cdot (k - k_0)^2$. With Lemma 4, let $\alpha = -\min_{f \in \mathcal{F}_{\text{1-Lip}}} J_D(f)$, then $J(k) = -k \cdot \alpha + \lambda \cdot (k - k_0)^2$, and $J(k)$ achieves its minimum when $k = \frac{\alpha}{2\lambda} + k_0$. When α goes to zero, *i.e.* p_{G} converges to p_{data}, the optimal k decreases. And when $p_{\mathrm{G}} = p_{\mathrm{data}}$, we have $\alpha = 0$ and optimal $k = k_0$. We choose $k_0 = 0$ in our experiments. The similar analysis applies to Lipschitz constraint based GANs and we use $\lambda \cdot k^2$ to enforcing k-Lipschitz for general Lipschitz constraint based GANs.

For practical methods, though spectral normalization Miyato et al. (2018) recently demonstrates their excellent results in training GANs, spectral normalization is an absolute constraint for Lipschitz over the entire space, i.e., constricting the maximum gradient of the entire space, which is unnecessary. On the other side, we also notice both penalty methods proposed in Gulrajani et al. (2017) and Petzka et al. (2017) are not the exactly implementing the Lipschitz continuity condition, because it not simply penalties the maximum gradient, but penalties all gradients towards 1 or penalties all these greater than one towards 1.

We found in our experiments that the existing methods including spectral normalization Miyato et al. (2018), gradient penalty Gulrajani et al. (2017), and Lipschitz penalty Petzka et al. (2017) all fail to converge to the optimal $f^*(x)$ in many of our synthetic experiments. We developed a new method for enforcing k-Lipschitz and we found in our experiments that the new method stably converges to the optimal $f^*(x)$.

The new method. Note that the practical methods of imposing k-Lipschitz is not the key contribution of this work, and it is far from well-validated. We plan a further work on this topic for a more rigorous study. But for the necessity for understanding our paper and reproducing of experiments, we introduce it as follows.

Combining the idea of spectral normalization and gradient penalty, we developed an new regularization for Lipschitz continuity in our experiments. Spectral normalization is actually constraining the maximum gradient over the entire space. And as we argued previously, enforcing Lipschitz continuity in the blending region is sufficient. Therefore, we propose to restricting the maximum gradient over the blending region:



(a) Gradient Penalty (b) Maximum Gradient Penalty

Figure 18: Comparison between gradient penalty and maximum gradient penalty, with P_r and P_g consist of **ten real and noise images**, respectively. The leftmost in each row is a $x \sim p_{\text{G}}$ and the second is its gradient $\frac{\partial f(x)}{\partial x}$. The interior are $x + \epsilon \cdot \frac{\partial f(x)}{\partial x}$ with increasing ϵ, which will pass through a real sample, and the rightmost is the corresponding $y \sim p_{\text{data}}$.

$$J_{\text{maxgp}} = \max_{\hat{x} \sim B(\mu,\nu)} [\left\|\nabla f(x)\right\|_2^2] \tag{18}$$

In practice, we sample $\hat{x}$ from training batch as in Gulrajani et al. (2017); Petzka et al. (2017). To improve the stability and reduce the biased introduced via batch sampling, we propose the keep track $\hat{x}$ with the maximum $\left\|\nabla f(x)\right\|_2$. A practical and light weight method is to maintain a list S_{max} that has the currently highest (top-k) $\left\|\nabla f(x)\right\|_2$ (initialized with random $\hat{x}$ samples), using the S_{max} as part of the batch estimation of J_{maxgp}, and update the S_{max} after each batch updating of the discriminator. In our experiment, S_{max} takes 1/2 batch, and the remaining 1/2 batch are random sampled. S_{max} always keeps track of the maximal 1/2 samples in the batch.

We comparing the practical result of gradient penalty $\boldsymbol{E}_{\hat{x} \sim B}[\left\|\nabla f(x)\right\|_2^2]$ and the proposed maximum gradient penalty in Figure 18. Before switching to maximum gradient penalty, we struggled for a long time and cannot achieve a high quality result as showed in Figure 18b. The other forms of gradient penalty Gulrajani et al. (2017); Petzka et al. (2017) perform similar as $\boldsymbol{E}_{\hat{x} \sim B}[\left\|\nabla f(x)\right\|_2^2]$.

F NO-DIFFERENTIABLE $f^*(x)$

If $f^*(x)$ is k-Lipschitz and $f^*(y) - f^*(x) = k \cdot d(x, y)$, we say that (x, y) are coupled. When a sample x is coupled with more than one y and these y lie in different directions of x, $f^*(x)$ is non-differentiable at x and it will has sub-gradient along each direction.

When the $f^*(x)$ non-differentiable, due to the smoothness of practically-used neural network, as we noticed in the experiments, it usually behaviors as that the gradient direction is pointing in the middle of these sub-gradient (more strictly, a linear combination of these sub-gradients).

It seems that when the p_{G} is discrete (simulating discrete token generation, such as language and music), it is easy to become non-differentiable: in the optimal transport perspective, once it is required to move to more than one targets, $f^*(x)$ is non-differentiable at this point.

One way to alleviate this above problem is adding noise (e.g. Gaussian) to each discrete token from p_{G}. The discrete token with different noises now disperse to different targets. In the practical generator for continuous token, such as images, this kind of non-differentiable problem naturally get solved.

The more serious non-differentiable problem requires trace back to the Monge problem Villani (2008), which theoretically discussed under which condition the optimal transport is a one-one mapping, which by nature solve the non-differentiable problem, as each sample now has a single target.

However, for the Monge problem is solvable, *i.e.* the mapping from p_{G} and p_{data} is one-one, it requires the $d(x, y)$ to be a strictly convex and super-linear Villani (2008). Unfortunately, the Euclidean distance, which is necessary to ensure the gradient direction from fake sample directly points toward real sample, does not fit this condition. So we currently does not figure out a practical solution to take advantage of the Monge problem related theories.

Nonetheless, even if $f^*(x)$ is non-differentiable, the gradient is also usually somehow pointing towards the real samples. And the empirical founding is that: when the p_{G} get close to p_{data}, the non-differentiable problem diminishes.

G PROOF OF THE THEOREM

Define $J_D = \boldsymbol{E}_{x\sim p_{\text{G}}}[\phi(f(x))] + \boldsymbol{E}_{x\sim p_{\text{data}}}[\varphi(f(x))] = \int p_{\text{G}}(x)\phi(f(x)) + p_{\text{data}}(x)\varphi(f(x))dx$. Note that $\partial_x J_D \triangleq \frac{\partial J_D}{\partial x} = p_{\text{G}}(x)\phi(f(x)) + p_{\text{data}}(x)\varphi(f(x))$. Define $J = J_D + \lambda \cdot k(f)^2$. Let $J_D^*(k) = \arg\min_{f\in\mathcal{F}_{\text{k-Lip}}} J_D$. Let $f^*(x) = \arg\min_f [J_D + \lambda \cdot k(f)^2]$, where $k(f)$ is Lipschitz constant of $f(x)$.

Lemma 1. $\forall x, \frac{\partial[\partial_x J_D]}{\partial f^*(x)} = 0$ if and only if $k(f^*) = 0$.

Proof.

(i) $\forall x, \frac{\partial[\partial_x J_D]}{\partial f^*(x)} = 0$ implies $k(f^*) = 0$.

For the optimal $f^*(x)$, it holds that $\frac{\partial J}{\partial k(f^*)} = \frac{\partial J_D^*}{\partial k(f^*)} + 2\lambda \cdot k(f^*) = 0$. $\forall x, \frac{\partial[\partial_x J_D]}{\partial f^*(x)} = 0$ implies $\frac{\partial J_D^*}{\partial k(f^*)} = 0$. We thus conclude that $k(f^*) = 0$.

(ii) $k(f^*) = 0$ implies $\forall x, \frac{\partial[\partial_x J_D]}{\partial f^*(x)} = 0$.

For the optimal $f^*(x)$, it holds that $\frac{\partial J}{\partial k(f^*)} = \frac{\partial J_D^*}{\partial k(f^*)} + 2\lambda \cdot k(f^*) = 0$. So $k(f^*) = 0$ implies $\frac{\partial J_D^*}{\partial k(f^*)} = 0$. $k(f^*) = 0$ also implies $\forall x, y, f^*(x) = f^*(y)$. If there exists some point x such that $\frac{\partial[\partial_x J_D]}{\partial f^*(x)} \neq 0$, then, given $\forall x, y, f^*(x) = f^*(y)$, it is obviously that $\frac{\partial J_D^*}{\partial k(f^*)} \neq 0$. It is contradictory to $\frac{\partial J_D^*}{\partial k(f^*)} = 0$. Thus we has $\forall x, \frac{\partial[\partial_x J_D]}{\partial f^*(x)} = 0$. □

Lemma 2. If $\forall x, y, f^*(x) = f^*(y)$, then $p_{\text{G}} = p_{\text{data}}$.

Proof.

$\forall x, y, f^*(x) = f^*(y)$ implies $k(f^*) = 0$. According to Lemma 1, $\forall x, \frac{\partial[\partial_x J_D]}{\partial f^*(x)} = p_{\text{G}}(x)\frac{\partial\phi(f^*(x))}{\partial f^*(x)} + p_{\text{data}}(x)\frac{\partial\varphi(f^*(x))}{\partial f^*(x)} = 0$. So $\frac{p_{\text{G}}(x)}{p_{\text{data}}(x)} = -\frac{\frac{\partial\varphi(f^*(x))}{\partial f^*(x)}}{\frac{\partial\phi(f^*(x))}{\partial f^*(x)}}$, and thus $\frac{p_{\text{G}}(x)}{p_{\text{data}}(x)}$ has a constant value, which straightforwardly implies $p_{\text{G}} = p_{\text{data}}$. □

Proof of Theorem 1.

(i) Considering the $f^*(x)$, $\forall x \in \bar{p}_{\text{G}} \cup \bar{p}_{\text{data}}$, if there does not exist a y such that $|f^*(y) - f^*(x)| = k(f^*) \cdot d(x, y)$, because $f^*(x)$ is the optimal, it must hold that $\frac{\partial[\partial_x J_D]}{\partial f^*(x)} = 0$. [4]

[4] Otherwise, as $f^*(x)$ is not constrained by the Lipschitz constraint, we can construct a better f^* by adjusting the value of $f^*(x)$ at x according to the non-zero gradient.

(ii) For $x \in \bar{p}_{\text{G}} \cup \bar{p}_{\text{data}} - \bar{p}_{\text{G}} \cap \bar{p}_{\text{data}}$, assuming $p_{\text{G}}(x) \neq 0$ and $p_{\text{data}}(x) = 0$, we have $\frac{\partial[\partial_x J_D]}{\partial f^*(x)} = p_{\text{G}}(x)\frac{\partial\phi(f^*(x))}{\partial f^*(x)} + p_{\text{data}}(x)\frac{\partial\varphi(f^*(x))}{\partial f^*(x)} = p_{\text{G}}(x)\frac{\partial\phi(f^*(x))}{\partial f^*(x)} > 0$, because $p_{\text{G}}(x) > 0$ and $\frac{\partial\phi(f^*(x))}{\partial f^*(x)} > 0$. Then, according to (i), there must exist a y such that $|f^*(y) - f^*(x)| = k(f^*) \cdot d(x, y)$. The other situation can be proved in the same way.

(iii) According to Lemma 2, in this situation that $p_{\text{G}} \neq p_{\text{data}}$, for the optimal $f^*(x)$, there must exist at least one pair of points x and y such that $x \neq y$ and $f^*(x) \neq f^*(y)$. If there are no x and y satisfying that $|f^*(y) - f^*(x)| = k(f^*) \cdot d(x, y)$, it will be contradictory to that $f^*(x)$ is optimal, because we can construct a better f^* by decreasing the value of $k(f)$ until there are two points, *e.g.* x and y, constrained by Lipschitz constraint, *i.e.* $|f^*(y) - f^*(x)| = k(f^*) \cdot d(x, y)$.

(iv) In Nash Equilibrium state, it holds that, for any $x \in \bar{p}_{\text{G}} \cup \bar{p}_{\text{data}}$, $\frac{\partial J}{\partial k(f)} = \frac{\partial J_D^*}{\partial k(f)} + 2\lambda \cdot k(f) = 0$ and $\frac{\partial[\partial_x J_D]}{\partial f(x)}\frac{\partial f(x)}{\partial x} = 0$. We claim that in the Nash Equilibrium state, the Lipschitz constant $k(f)$ must be 0. If $k(f) \neq 0$, according to Lemma 1, there must exist a point $\hat{x}$ such that $\frac{\partial[\partial_{\hat{x}} J_D]}{\partial f(\hat{x})} \neq 0$. And according to (i), it must hold that $\exists \hat{y}$ fitting $|f(\hat{y}) - f(\hat{x})| = k(f) \cdot d(\hat{x}, \hat{y})$. According to Theorem 3, we have $\left\|\frac{\partial f(x)}{\partial x}\big|_{x=\hat{x}}\right\|_2 = k(f) \neq 0$. This is contradictory to that $\frac{\partial[\partial_x J_D]}{\partial f(x)}\frac{\partial f(x)}{\partial x}\big|_{x=\hat{x}} = 0$. Thus $k(f) = 0$, that is, $\forall x \in \bar{p}_{\text{G}} \cup \bar{p}_{\text{data}}$, $\frac{\partial f(x)}{\partial x} = 0$, which means $\forall x, y, f(x) = f(y)$. According to Lemma 2, $\forall x, y, f(x) = f(y)$ implies $p_{\text{G}} = p_{\text{data}}$. Thus $p_{\text{G}} = p_{\text{data}}$ is the only Nash Equilibrium of our system. □

Remark: For the Wasserstein distance, $\frac{\partial[\partial_x J_D]}{\partial f^*(x)} = 0$ if and only if $p_{\text{G}}(x) = p_{\text{data}}(x)$. For the Wasserstein distance, enforcing a dynamic k-Lipschitz also benefits: at the convergence state, it holds $\frac{\partial f^*(x)}{\partial x} = 0$.

The proof of Theorem 3 can be found at Section D.

H THE IMPORTANCE OF EQUATION 12

Requiring $\phi(x)$ and $\varphi(x)$ to satisfy Equation 12 is important, because it is the non-trivial condition that makes sure $\boldsymbol{E}_{x\sim p_{\text{G}}}[\phi(f(x))] + \boldsymbol{E}_{x\sim p_{\text{data}}}[\varphi(f(x))] + \lambda \cdot k(f)^2$ has global minimum with respect to f.

Lemma 3. If $\phi(x)$ and $\varphi(x)$ satisfies Equation 12, then for any fixed p_{G} and p_{data}, $\boldsymbol{E}_{x\sim p_{\text{G}}}[\phi(f(x))] + \boldsymbol{E}_{x\sim p_{\text{data}}}[\varphi(f(x))]$ has global minimum with respect to f, for any fixed $k(f) = \hat{k}$.

Proof.

(i) Assuming p_{G} and p_{data} are two delta distributions.

Given p_{G} and p_{data} are two delta distributions, according to Theorem 1 and Theorem 2, for $x \sim p_{\text{G}}$ and $y \sim p_{\text{data}}$, $f^*(y) - f^*(x) = \hat{k} \cdot d(x, y)$. Let $f^*(x) = \alpha$ and $\beta = \hat{k} \cdot d(x, y)$, then $f^*(y) = \alpha + \beta$. Define $J_D(\alpha) = \boldsymbol{E}_{x\sim p_{\text{G}}}[\phi(f(x))] + \boldsymbol{E}_{x\sim p_{\text{data}}}[\varphi(f(x))] = \phi(\alpha) + \varphi(\alpha + \beta)$.

Given $\exists\, a$ such that $\phi'(a) + \varphi'(a) = 0$, $\frac{\partial[}{\partial 2}]\phi(x)x \geq 0$ and $\frac{\partial[}{\partial 2}]\varphi(x)x \geq 0$, we have, when α is small enough (such that, $\alpha < a$ and $\alpha + \beta < a$), $J_D'(\alpha) = \phi'(\alpha) + \varphi'(\alpha + \beta) \leq \phi'(a) + \varphi'(a) = 0$. Similarly, when α is large enough (such that, $\alpha > a$ and $\alpha + \beta > a$), $J_D'(\alpha) = \phi'(\alpha) + \varphi'(\alpha + \beta) \geq \phi'(a) + \varphi'(a) = 0$.

Therefore, $J_D(\alpha)$ is convex with respect to α and there exists an α_0 such that $J_D'(\alpha_0) = 0$, where J_D achieves its the global minimum. When $\frac{\partial[}{\partial 2}]\phi(x)x > 0$ and $\frac{\partial[}{\partial 2}]\varphi(x)x > 0$, it is the unique global minimum.

(ii) The general case.

For any p_{G} and p_{data}, given any initial $f(x)$, we construct a $f^*(x)$ in the following manner.

For any point $x \in \bar{p}_{\text{G}} \cup \bar{p}_{\text{data}}$, we define the initial bounding set $S_x = \{x\}$. We will use S_x to keep track of all points that have direct or indirect bounding relationship with x. When we say x and y has bounding relationship, we mean $|f^*(y) - f^*(x)| = k(f^*) \cdot d(x, y)$. Define $\nabla_x = \int_{t \in S_x} p_{\text{G}}(t) \frac{\partial \phi(f(t))}{\partial f(t)} + p_{\text{data}}(t) \frac{\partial \varphi(f(t))}{\partial f(t)} dt$.

We apply the following process separately and sequentially to all $x \in \bar{p}_{\text{data}}$. For every point $x \in \bar{p}_{\text{data}}$, we define the initial temporal bounding set $M_x = S_x$. We will gradually adjust the $f(x)$ for all points in M_x with the same amount according to ∇_x, and iteratively collect all sets that have bounding relationship with any point in M_x into M_x, *i.e.*, $M_x := M_x + S_y, \forall y$ that holds $\exists x \in M_x$ such that $|f^*(y) - f^*(x)| = k(f^*) \cdot d(x, y)$, at the same time, we update $S_x := S_x + S_y, \forall y \in \bar{p}_{\text{G}}$ that holds $\exists x \in S_x$ such that $f^*(x) - f^*(y) = k(f^*) \cdot d(x, y)$. For any $x \in \bar{p}_{\text{data}}$, we will continue the increasing of the $f(x)$ for all x in M_x until $\nabla_x = 0$.

After that, we apply a similar process for each $x \in \bar{p}_{\text{G}}$. For every $x \in \bar{p}_{\text{G}}$, we would gradually adjust the $f(x)$ for all points in M_x with the same amount according to ∇_x if $\nabla_x \neq 0$, until $\nabla_x = 0$. During the process, we iteratively collect all sets that have bounding relationship with any point in M_x into M_x, *i.e.*, $M_x := M_x + S_y, \forall y$ that holds $\exists x \in M_x$ such that $|f^*(y) - f^*(x)| = k(f^*) \cdot d(x, y)$, at the same time, we update $S_x := S_x + S_y, \forall y \in \bar{p}_{\text{data}}$ that holds $\exists x \in S_x$ such that $f^*(y) - f^*(x) = k(f^*) \cdot d(x, y)$.

It can be proved that the "continue..until..." process would finally end for any $x \in \bar{p}_{\text{G}} \cup \bar{p}_{\text{data}}$. If not, then M_x will collect all $x \in \bar{p}_{\text{G}} \cup \bar{p}_{\text{data}}$ at the ending. Given $\exists\, a$, $\frac{\partial \phi(x)}{\partial x}\big|_{x=a} + \frac{\partial \varphi(x)}{\partial x}\big|_{x=a} = 0$ and $\frac{\partial[}{\partial 2}]\phi(x)x \geq 0$ and $\frac{\partial[}{\partial 2}]\varphi(x)x \geq 0$, and $M_x = \bar{p}_{\text{G}} \cup \bar{p}_{\text{data}}$: if $\forall x \in M_x, f(x) < a$, it would holds that $\nabla_x \geq 0$; if $\forall x \in M_x, f(x) < a$, it would holds that $\nabla_x \leq 0$, and there exists a $f(x)$ such that $\nabla_x = 0$.

When all the processes ended, it can be proved that for any point $x \in \bar{p}_{\text{G}} \cup \bar{p}_{\text{data}}$, it hold $\nabla_s = 0$, and $f(x)$ achieves its minimum. As showed in the above paragraph, the $\int_{t \in S_x} p_{\text{G}}(t)\phi(f(t)) + p_{\text{data}}(t)\varphi(f(t))dt$ in construction process is convex all along, so the minimum is the global minimum. And when $\frac{\partial[}{\partial 2}]\phi(x)x > 0$ and $\frac{\partial[}{\partial 2}]\varphi(x)x > 0$, it is strictly convex and thus the global minimum is the unique global minimum. □

Lemma 4. If $\phi(x)$ and $\varphi(x)$ satisfies Equation 12, then for any fixed p_{G} and p_{data}, $\boldsymbol{E}_{x \sim p_{\text{G}}}[\phi(f(x))] + \boldsymbol{E}_{x \sim p_{\text{data}}}[\varphi(f(x))]$ monotonically increases as $k(f)$ decreases.

Proof.

(i) Assuming p_{G} and p_{data} are two delta distributions

Given p_{G} and p_{data} are two delta distributions, according to Theorem 1 and Theorem 2, for $x \sim p_{\text{G}}$ and $y \sim p_{\text{data}}$, $f^*(y) - f^*(x) = k \cdot d(x, y)$. Let $f^*(x) = \alpha$ and $\beta = k \cdot d(x, y) > 0$, then $f^*(y) = \alpha + \beta$. Define $J_D(\beta) = \min_{f \in \mathcal{F}_{\text{k-Lip}}} \boldsymbol{E}_{x \sim p_{\text{G}}}[\phi(f(x))] + \boldsymbol{E}_{x \sim p_{\text{data}}}[\varphi(f(x))] = \min_\alpha \phi(\alpha) + \varphi(\alpha + \beta)$. We need to prove $J_D(\beta)$ is monotonically decreasing, for $\beta \geq 0$.

Let $0 \leq \beta_1 < \beta_2$, let $\alpha_1 = \min_\alpha \phi(\alpha) + \varphi(\alpha + \beta_1)$ and $\alpha_2 = \min_\alpha \phi(\alpha) + \varphi(\alpha + \beta_2)$. Given $\frac{\partial \varphi(x)}{\partial x} < 0$ and $\beta_1 < \beta_2$, we have $\phi(\alpha_1) + \varphi(\alpha_1 + \beta_1) > \phi(\alpha_1) + \varphi(\alpha_1 + \beta_2)$. Given $\alpha_2 = \min_\alpha \phi(\alpha) + \varphi(\alpha + \beta_2)$, we further have $\phi(\alpha_1) + \varphi(\alpha_1 + \beta_1) > \phi(\alpha_1) + \varphi(\alpha_1 + \beta_2) \geq \phi(\alpha_2) + \varphi(\alpha_2 + \beta_2)$. Done.

Additionally, with $\frac{\partial \phi(x)}{\partial x}\big|_{x=\alpha_1} + \frac{\partial \varphi(x)}{\partial x}\big|_{x=\alpha_1+\beta_1} = 0$ and $\frac{\partial[}{\partial 2}]\varphi(x)x \geq 0$, we have $\frac{\partial \phi(x)}{\partial x}\big|_{x=\alpha_1} + \frac{\partial \varphi(x)}{\partial x}\big|_{x=\alpha_1+\beta_2} \geq 0$. Providing $\frac{\partial \phi(x)}{\partial x}\big|_{x=\alpha_2} + \frac{\partial \varphi(x)}{\partial x}\big|_{x=\alpha_2+\beta_2} = 0$, $\frac{\partial[}{\partial 2}]\phi(x)x \geq 0$ and $\frac{\partial[}{\partial 2}]\varphi(x)x \geq 0$, we get $\alpha_2 \leq \alpha_1$. That is, $\alpha_2 \leq \alpha_1 < \beta_1 < \beta_2$. When $\frac{\partial[}{\partial 2}]\phi(x)x > 0$ and $\frac{\partial[}{\partial 2}]\varphi(x)x > 0$, we have $\alpha_2 < \alpha_1 < \beta_1 < \beta_2$.

□

I HYPER-PARAMETER & NETWORK ARCHITECTURE

We follow the network architecture proposed in Gulrajani et al. (2017) to conduct our experiments on Cifar-10, Tiny Imagenet, Oxford 102. The details of network architecture are in Table 3.

Generator:

Operation	Kernel	Resample	Output Dims
Noise	N/A	N/A	128
Linear	N/A	N/A	$128\times4\times4$
Residual block	3×3	UP	$128\times8\times8$
Residual block	3×3	UP	$128\times16\times16$
Residual block	3×3	UP	$128\times32\times32$
Conv,tanh	3×3	N/A	$3\times32\times32$

Critic:

Operation	Kernel	Resample	Output Dims
Residual Block	$3\times3\times2$	Down	$128\times16\times16$
Residual Block	$3\times3\times2$	Down	$128\times8\times8$
Residual Block	$3\times3\times2$	N/A	$128\times8\times8$
Residual Block	$3\times3\times2$	N/A	$128\times8\times8$
ReLU,mean pool	N/A	N/A	128
Linear	N/A	N/A	1

Optimizer: Adam with beta1=0.0, beta2=0.9;
For more details please refer to our published codes.

Table 3: Hyper-parameter and Network Architectures