GRADIENT PENALTY FROM A MAXIMUM MARGIN PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

A popular heuristic for improved performance in Generative adversarial networks (GANs) is to use some form of gradient penalty on the discriminator. This gradient penalty was originally motivated by a Wasserstein distance formulation. However, the use of gradient penalty in other GAN formulations is not well motivated. We present a unifying framework of expected margin maximization and show that a wide range of gradient-penalized GANs (e.g., Wasserstein, Standard, Least-Squares, and Hinge GANs) can be derived from this framework. Our results imply that employing gradient penalties induces a large-margin classifier (thus, a large-margin discriminator in GANs). We describe how expected margin maximization helps reduce vanishing gradients at fake (generated) samples, a known problem in GANs. From this framework, we derive a new L^∞ gradient norm penalty with Hinge loss which generally produces equally good (or better) generated output in GANs than L^2-norm penalties (based on the Fréchet Inception Distance).

1 INTRODUCTION

Generative adversarial networks (GANs) (Goodfellow et al., 2014) are a very successful class of generative models. Their most common formulation involves a game played between two competing neural networks, the discriminator D and the generator G. D is a classifier trained to distinguish real from fake examples, while G is trained to generate fake examples that will confuse D into recognizing them as real. When the discriminator's objective is maximized, it yields the value of a specific divergence (i.e., a distance between probability distributions) between the distributions of real and fake examples. The generator then aims to minimize that divergence (although this interpretation is not perfect; see Jolicoeur-Martineau (2018a)).

Importantly, many GANs apply some form of gradient norm penalty to the discriminator (Gulrajani et al., 2017; Fedus et al., 2017b; Mescheder et al., 2018; Karras et al., 2019). Gradient norm penalty has been widely adopted by the GAN community as a useful heuristic to improve the stability of GANs and the quality of the generated outputs. This penalty was originally motivated by a Wasserstein distance formulation in Gulrajani et al. (2017). However, its use in other GAN formulations is not well motivated. Given its success, one might wonder how one could derive an arbitrary GAN formulation with a gradient penalty?

In this paper, we derive a framework which shows that gradient penalty arises in GANs from using a maximum margin classifier as discriminator. We then use this framework to better understand GANs and devise better gradient penalties.

The main contributions of this paper are:

1. A unifying framework of expected margin maximization and showing that gradient-penalized versions of most discriminator/classifier loss functions (Wasserstein, Cross-entropy, Least-Squares, Hinge-Loss) can be derived from this framework.
2. A new method derived from our framework, a L^∞ gradient norm penalty with Hinge function. We hypothesize and show that this method works well in GAN.
3. We describe how margin maximization (and thereby gradient penalties) helps reduce vanishing gradients at fake (generated) samples, a known problem in many GANs.

4. We derive the margins of Relativistic paired and average GANs (Jolicoeur-Martineau, 2018b; 2019).

The paper is organized as follows. In Section 2, we show how gradient penalty arises from the Wasserstein distance in the GAN literature. In Section 3, we explain the concept behind maximum-margin classifiers (MMCs) and how they lead to some form of gradient penalty. In Section 4, we present our generalized framework of maximum-margin classification and experimentally validate it. In Section 5, we discuss of the implications of this framework on GANs and hypothesize that L^1-norm margins may lead to more robust classifiers. Finally, in Section 6, we provide experiments to test the different GANs resulting from our framework. Note that due to space constraints, we relegated the derivations of the margins of Relativistic GANs to Appendix C.

2 Gradient Penalty from the GAN Literature

2.1 Notation

We focus on binary classifiers. Let f be the classifier and $(x, y) \sim \mathbb{D}$ the distribution (of a dataset D) with n data samples x and labels y. As per SVM literature, $y = 1$ when x is sampled from class 1 and $y = -1$ when x is sampled from class 2. Furthermore, we denote $x_1 = x|(y = 1) \sim \mathbb{P}$ and $x_2 = x|(y = -1) \sim \mathbb{Q}$ as the data samples from class 1 and class 2 respectively (with distributions $\mathbb{P}$ and $\mathbb{Q}$). When discussing GANs, $x_1 \sim \mathbb{P}$ (class 1) refer to real data samples and $x_2 \sim \mathbb{Q}$ (class 2) refer to fake data samples (produced by the generator). The L^∞-norm is defined as: $||x||_\infty = \max(|x_1|, |x_2|, \dots, |x_k|)$.

The critic (C) is the discriminator (D) before applying any activation function (i.e., $D(x) = a(C(x))$, where a is the activation function). For consistency with existing literature, we will generally refer to the critic rather than the discriminator.

2.2 GANs

GANs can be formulated in the following way:

$$\max_{C:\mathcal{X}\to\mathbb{R}} \mathbb{E}_{x_1\sim\mathbb{P}}\left[f_1(C(x_1))\right] + \mathbb{E}_{z\sim\mathbb{Z}}\left[f_2(C(G(z)))\right], \tag{1}$$

$$\min_{G:Z\to\mathcal{X}} \mathbb{E}_{z\sim\mathbb{Z}}\left[f_3(C(G(z)))\right], \tag{2}$$

where $f_1, f_2, f_3 : \mathbb{R} \to \mathbb{R}$, $\mathbb{P}$ is the distribution of real data with support $\mathcal{X}$, $\mathbb{Z}$ is a multivariate normal distribution with support $Z = \mathbb{R}$, $C(x)$ is the critic evaluated at x, $G(z)$ is the generator evaluated at z, and $G(z) \sim \mathbb{Q}$, where $\mathbb{Q}$ is the distribution of fake data.

Many variants exist; to name a few: Standard GAN (SGAN) (Goodfellow et al., 2014) corresponds to $f_1(z) = \log(\text{sigmoid}(z))$, $f_2(z) = \log(\text{sigmoid}(-z))$, and $f_3(z) = -f_1(z)$. Least-Squares GAN (LSGAN) (Mao et al., 2017) corresponds to $f_1(z) = -(1-z)^2$, $f_2(z) = -(1+z)^2$, and $f_3(z) = -f_1(z)$. HingeGAN (Lim & Ye, 2017) corresponds to $f_1(z) = -max(0, 1-z)$, $f_2(z) = -max(0, 1+z)$, and $f_3(z) = -z$.

2.3 Integral Probability Metric Based GANs

An important class of statistical divergences (distances between probability distributions) are Integral probability metrics (IPMs) (Müller, 1997). IPMs are defined in the following way:

$$IPM_{\mathcal{F}}(\mathbb{P}||\mathbb{Q}) = \sup_{C\in\mathcal{F}} \mathbb{E}_{x_1\sim\mathbb{P}}[C(x_1)] - \mathbb{E}_{x_2\sim\mathbb{Q}}[C(x_2)],$$

where $\mathcal{F}$ is a class of real-valued functions.

A widely used IPM is the Wasserstein's distance (W_1), which focuses on the class of all 1-Lipschitz functions. This corresponds to the set of functions C such that $\frac{C(x_1)-C(x_2)}{d(x_1,x_2)} \leq 1$ for all x_1,x_2, where $d(x_1, x_2)$ is a metric. W_1 also has a primal form which can be written in the following way:

$$W_1(\mathbb{P}, \mathbb{Q}) := \inf_{\pi\in\Pi(\mathbb{P},\mathbb{Q})} \int_{M\times M} d(x_1, x_2)\, \mathrm{d}\pi(x_1, x_2),$$

where $\Pi(\mathbb{P}, \mathbb{Q})$ is the set of all distributions with marginals $\mathbb{P}$ and $\mathbb{Q}$ and we call π a coupling.

The Wasserstein distance has been highly popular in GANs due to the fact that it provides good gradient for the generator in GANs which allows more stable training.

IPM-based GANs (Arjovsky et al., 2017; Gulrajani et al., 2017) attempt to solve the following problem

$$\min_G \max_{C \in \mathcal{F}} \mathbb{E}_{x_2 \sim \mathbb{P}}[C(x_1)] - \mathbb{E}_{z \sim \mathbb{Z}}[C(G(z))].$$

2.4 Gradient Penalty as a way to estimate the Wasserstein Distance

To estimate the Wasserstein distance using its dual form (as a IPM), one need to enforce the 1-Lipschitz property on the critic. Gulrajani et al. (2017) showed that one could impose a gradient penalty, rather than clamping the weights as originally done (Arjovsky et al., 2017), and that this led to better GANs. More specifically, they showed that if the optimal critic $f^*(x)$ is differentiable everywhere and that $\hat{x} = \alpha x_1 + (1-\alpha)x_2$ for $0 \leq \alpha \leq 1$, we have that $||\nabla C^*(\hat{x})||_2 = 1$ almost everywhere for all pair (x_1, x_2) which comes from the optimal coupling π^*.

Sampling from the optimal coupling is difficult so they suggested to softly penalize $\mathbb{E}_{\tilde{x}}(||\nabla_{\tilde{x}} C(\tilde{x})||_2 - 1)^2$, where $\tilde{x} = \alpha x_1 + (1-\alpha)x_2$, $\alpha \sim U(0,1)$, $x_1 \sim \mathbb{P}$, and $x_2 \sim \mathbb{Q}$. They called this approach Wasserstein GAN with gradient-penalty (WGAN-GP). However, note that this approach does not necessarily estimate the Wasserstein distance since we are not sampling from π^* and f^* does not need to be differentiable everywhere (Petzka et al., 2017).

Of importance, gradient norm penalties of the form $\mathbb{E}_x(||\nabla_x D(x)||_2 - \delta)^2$, for some $\delta \in \mathbb{R}$ are very popular in GANs. Remember that $D(x) = a(C(x))$; in the case of IPM-based-GANs, we have that $D(x) = C(x)$. It has been shown that the GP-1 penalty ($\delta = 1$), as in WGAN-GP, also improves the performance of non-IPM-based GANs (Fedus et al., 2017a). Another successful variant is GP-0 ($\delta = 0$ and $x \sim \mathbb{P}$) (Mescheder et al., 2018; Karras et al., 2019). Although there are explanations to why gradient penalties may be helpful (Mescheder et al., 2018; Kodali et al., 2017; Gulrajani et al., 2017), the theory is still lacking.

3 Maximum-Margin Classifiers

In this section, we define the concepts behind maximum-margin classifiers (MMCs) and show how it leads to a gradient penalty.

3.1 Decision Boundary and Margin

The *decision boundary* of a classifier is defined as the set of points x_0 such that $f(x_0) = 0$.

The margin is either defined as i) the minimum distance between a sample and the boundary, or ii) the minimum distance between the *closest sample* to the boundary and the boundary. The former thus corresponds to the *margin of a sample* and the latter corresponds to the *margin of a dataset* . In order to disambiguate the two cases, we refer to the former as the *margin* and the latter as the *minimum margin*.

3.2 Geometric Margin and Gradient Penalty

The first step towards obtaining a MMC is to define the L^p-norm margin:

$$\gamma(x) = \min_{x_0} ||x_0 - x||_p \quad \text{s.t.} \quad f(x_0) = 0 \tag{3}$$

With a linear classifier (i.e., $f(x) = w^T x$), there is a close form solution. However, the formulation of the L^p-norm margin equation 3 has no closed form for arbitrary non-linear classifiers. A way to derive an approximation of the margin is to use Taylor's approximation before solving the problem

(as done by Matyasko & Chau (2017) and Elsayed et al. (2018)):

$$\begin{aligned}\gamma_p(x) &= \min_r ||r||_p \quad \text{s.t.} \quad f(x+r) = 0 \\ &\approx \min_r ||r||_p \quad \text{s.t.} \quad f(x) + \nabla_x f(x)^T r = 0 \\ &= \frac{|f(x)|}{||\nabla_x f(x)||_q},\end{aligned}$$

where $||\cdot||_q$ is the dual norm (Boyd & Vandenberghe, 2004) of $||\cdot||_p$. By Hölder's inequality (Hölder, 1889; Rogers, 1888), we have that $1/p + 1/q = 1$. This means that if $p = 2$, we still get $q = 2$; if $p = \infty$, we get $q = 1$; if $p = 1$, we get $q = \infty$.

The goal of MMCs is to maximize a margin, but also to obtain a classifier. To do so, we simply replace $\alpha(x) = |f(x)|$ by $\widetilde{\alpha}(x, y) = yf(x)$. We call $\widetilde{\alpha}$ the *functional margin*. After replacement, we obtain the *geometric margin*:

$$\widetilde{\gamma}(x, y) = \frac{yf(x)}{||\nabla_x f(x)||_q}$$

If $p = 2$ and $f(x)$ is linear, this leads to the same geometric margin used in Support Vector-Machines (SVMs). Matyasko & Chau (2017) used this result to generalize Soft-SVMs to arbitrary classifiers by simply penalizing the L^p-norm of the gradient rather than penalizing the L^p-norm of the model's weights (as done in SVMs). Meanwhile, Elsayed et al. (2018) used this result in a multi-class setting and maximized the geometric margin directly.

4 Generalized framework of Maximum-margin classification

4.1 Framework

Here we show how to generalize the idea behind maximizing the geometric margin into arbitrary loss functions with a gradient penalty. Directly maximizing the geometric margin is an ill-posed problem. The numerator and denominator are dependent on one another; increasing the functional margin also increases the norm of the gradient (and vice-versa). Thereby, there are infinite solutions which maximize the geometric margin. For this reason, the common approach (as in SVM literature; see Cortes & Vapnik (1995)) is to: i) constrain the numerator and minimize the denominator, or ii) constrain the denominator and maximize the numerator.

Approach i) consists of minimizing the denominator and constraining the numerator using the following formulation:

$$\min_f ||\nabla_x f(x)||_p \quad \text{s.t.} \quad yf(x) \geq 1 \, \forall \, (x, y) \in D \tag{4}$$

The main limitation of this approach is that it only works when the data are separable. However, if we take the opposite approach of maximizing a function of $yf(x)$ and constraining the denominator $||f(x)||_2$, we can still solve the problem with non-separable data. This corresponds to approach ii):

$$\max_f \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[yf(x)\right] \text{ s.t. } ||\nabla_x f(x)||_q \leq 1 \text{ or } ||\nabla_x f(x)||_q = 1. \tag{5}$$

The constraint chosen can be enforced by either i) using a KKT multiplier (Kuhn & Tucker, 1951; Karush, 1939) or ii) approximately imposing it with a soft-penalty. Furthermore, one can use any margin-based loss function rather than directly maximize $yf(x)$. Thus, we can generalize this idea by using the following formulation:

$$\min_f \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[L(yf(x)) + \lambda g(||\nabla_x f(x)||_q)\right]. \tag{6}$$

where $L, g : \mathbb{R} \to \mathbb{R}$ and λ is a scalar penalty term. There are many potential choices of L and g which we can use.

If L is chosen to be the hinge function (i.e., $L(z) = max(0, 1 - z)$), we ignore samples far from the boundary (as in Hard-Margin SVMs). For general choices of L, every sample may influence

the solution. The identity function $L(z) = z$, cross entropy with sigmoid activation $L(z) = -\log(\text{sigmoid}(z))$ and least-squares $L(z) = (1-z)^2$ are also valid choices.

A standard choice of g is $g(z) = (z^2 - 1)$. This corresponds to constraining $||\nabla_x f(x)||_2^2 = 1$ or $||\nabla_x f(x)||_2^2 \leq 1$ for all x (by KKT conditions). As an alternative, we can also consider soft constraints of the form $g(z) = (z-1)^2$ or $g(z) = max(0, z-1)$. The first function enforces a soft equality constraint so that $z \approx 1$ while the second function enforces a soft inequality constraint so that $z \leq 1$. Soft constraints are useful if the goal is not to obtain the maximum margin solution but to obtain a solution that leads to a large-enough margin.

Of importance, MMCs can be seen as a generalization of Support Vector Machines (SVMs). When $p = 2$ and f is linear ($f(x) = w^T x$), equation 4 corresponds exactly to Hard-Margin SVMs and equation 6 with $L(z) = max(0, 1-z)$ and $g(z) = (z^2 - 1)$ corresponds exactly to Soft-Margin SVMs (Cortes & Vapnik, 1995).

4.2 Experimental evidence of large margin from gradient penalties

We ran experiments to empirically show that gradient-penalized classifiers (trained to optimize equation 6) maximize the expected margin. We used the swiss-roll dataset (Marsland, 2015) to obtain two classes (one is the swiss-roll and one is the swiss-roll scaled by 1.5). The results are shown in Table 1 (Details of the experiments are in Appendix A).

Table 1: Expected L^p Margin for different types of gradient penalties (or none). Classifier was trained on the swiss-roll dataset with a cross-entropy loss function.

Type of gradient penalty	Expected L^p Margin $p=2$	$p=1$	$p=\infty$
No gradient penalty	.27	.25	.24
$g(z) = (z-1)^2$			
L^2 gradient penalty (L^2 margin)	.62	.75	.64
L^∞ gradient penalty (L^1 margin)	.43	.58	.33
L^1 gradient penalty (L^∞ margin)	.69	.85	.60
$g(z) = \max(0, z-1)$			
L^2 gradient penalty (L^2 margin)	.43	.53	.37
L^∞ gradient penalty (L^1 margin)	.41	.56	.31
L^1 gradient penalty (L^∞ margin)	.43	.44	.42

We observe that we obtain much larger expected margins (generally 2 to 3 times bigger) when using a gradient penalty; this is true for all types of gradient penalties.

5 Implications of the Maximum Margin Framework on GANs

5.1 GANs can be derived from the MMC Framework

Although not immediately clear given the different notations, let $f(x) = C(x)$ and we have:

$$\mathbb{E}_{(x,y)\sim\mathbb{D}}\left[L(yf(x))\right] = \mathbb{E}_{x_1\sim\mathbb{P}}[L(C(x_1))] + \mathbb{E}_{z\sim\mathbb{Z}}[L(-C(G(z)))].$$

Thus, the objective functions of the discriminator/critic in many penalized GANs are equivalent to the ones from MMCs based on equation 6. We also have that $L(z) = \log(\text{sigmoid}(z))$ corresponds to SGAN, $L(z) = (1-z)^2$ corresponds to LSGAN, and $L(z) = max(0, 1-z)$ corresponds to HingeGAN. When $g(z) = (z-1)^2$, we also have that $L(z) = z$ corresponds to WGAN-GP. Thus, most L^p-norm gradient penalized GANs imply that the discriminator approximately maximize an expected L^q-norm margin.

5.2 WHY DO MAXIMUM-MARGIN CLASSIFIERS MAKE GOOD GAN DISCRIMINATORS/CRITICS?

To show that maximizing an expected margin leads to better GANs, we prove the following statements:

1. classifier maximizes an expected margin $\iff$ classifier has a fixed Lipschitz constant
2. MMC with a fixed Lipschitz constant $\implies$ better gradients at fake samples
3. better gradients at fake samples $\implies$ stable GAN training.

5.2.1 EQUIVALENCE BETWEEN GRADIENT NORM CONSTRAINTS AND LIPSCHITZ FUNCTIONS

As stated in Section 2.4, the WGAN-GP approach of softly enforcing $||\nabla_{\tilde{x}} f(\tilde{x})||_2 \approx 1$ at all interpolations between real and fake samples does not ensure that we estimate the Wasserstein distance (W_1). On the other hand, we show here that enforcing $||\nabla_x f(x)||_2 \leq 1$ is sufficient in order to estimate W_1.

Assuming $d(x_1, x_2)$ is a L^p-norm, $p \geq 2$ and $f(x)$ is differentiable, we have that:

$$||\nabla f(x)||_p \leq K \iff f \text{ is K-Lipschitz on } L^p.$$

See appendix for the proof. Adler & Lunz (2018) showed a similar result on dual norms.

This suggests that, in order to work on the set of Lipschitz functions, we should enforce that $||\nabla_x f(x)|| \leq 1$ for all x. This can be done, through equation 6, by choosing $g(z) = (z^2 - 1)$ or, in approximation (using a soft-constraint), by choosing $g(z) = \max(0, z - 1)$. Petzka et al. (2017) suggested using a similar function (the square hinge) in order to only penalize gradient norms above 1.

If we let $L(z) = z$ and $g(z) = \max(0, z - 1)$, we have an IPM over all Lipschitz functions. thus, we effectively approximate W_1. This means that W_1 can be found through maximizing a geometric margin. Meanwhile, WGAN-GP only leads to a lower bound on W_1.

Importantly, most successful GANs (Brock et al., 2018; Karras et al., 2019; 2017) either enforce the 1-Lipschitz property using Spectral normalization (Miyato et al., 2018) or use some form of gradient norm penalty (Gulrajani et al., 2017; Mescheder et al., 2018). Since 1-Lipschitz is equivalent to enforcing a gradient norm constraint (as shown above), we have that most successful GANs effectively train a discriminator/critic to maximize a geometric margin.

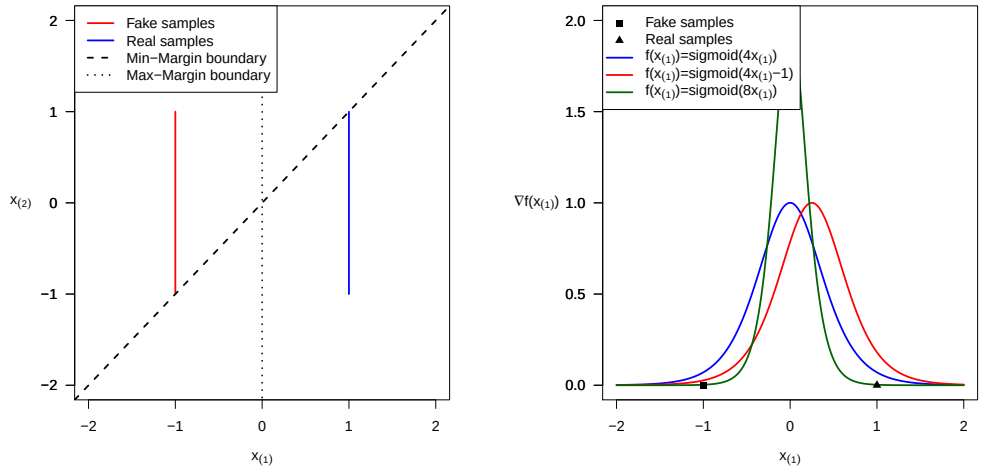
The above shows that training an MMC based on equation (6) is equivalent to training a classifier with a fixed Lipschitz constant.

5.2.2 MMC LEADS TO BETTER GRADIENT AT FAKE SAMPLES

Consider a simple two-dimensional example where $x = (x_{(1)}, x_{(2)})$. Let real data (class 1) be uniformly distributed on the line between $(1, -1)$ and $(1, 1)$. Let fake data (class 2) be uniformly distributed on the line between $(-1, -1)$ and $(-1, 1)$. This is represented by Figure 1a. Clearly, the maximum-margin boundary is the line $x_{(1)} = 0$ and any classifier should learn to ignore $x_{(2)}$.

Consider a non-linear classifier of the form $f(x) = \text{sigmoid}(w_1 x_{(1)} + w_0)$ (See Figure 1b). To ensure we obtain an MMC, we need to enforce $||\nabla_x f(x)|| \leq K$.

The best classifier with Lipschitz constant $K = 1$ is obtained by choosing $w_1 = 4$. The maximum-margin boundary is at $x_{(1)} = 0$ (which we get by taking $w_0 = 0$; blue curve in Figure 1b); for this choice, we have that $f(x_r) = .02$ and $f(x_f) = .98$ for real (x_r) and fake (x_f) samples respectively. Meanwhile, if we take a slightly worse margin with boundary at $x_{(1)} = \frac{1}{4}$ (equivalent to choosing $w_0 = -1$; red curve in Figure 1b), we have that $f(x_r) = .01$ and $f(x_f) = .95$ for real and fake samples respectively. Thus, both solutions almost perfectly classify the samples. However, the optimal margin has gradient .07, while the worse margin has gradient .03 at fake samples; this is why maximizing a margin lead to similar signal for real and fake samples. Furthermore, if we had enforced a bigger Lipschitz constant ($K = 2$), the best classifier would have been obtained with $w_1 = 8$ (green curve in Figure 1b); this would have caused vanishing gradients at fake samples unless we had scaled up the learning up. Thus, for a fixed or decreasing learning rate, it is important to fix K (and ideally to a small value) in order for the gradient signal to be strong at fake samples.



(a) (b)

Figure 1: a) Two-dimensional GAN example with different choices of boundaries, b) $\nabla f(x_{(1)})$ at different values of $x_{(1)}$ for the two-dimensional example assuming a sigmoid function.

In summary, the maximum-margin discriminator provides a stronger signal at fake samples by preventing a sharp change in the discriminator (i.e., small gradient near real/fake data and large gradient between real and fake data) and centering the classifier so that the gradients at real and fake samples are similar. This further suggests that imposing a gradient penalty in the interpolation between real and fake data (as done in WGAN-GP) is most sensible to ensure that the gradient norm remains small between real and fake data.

5.2.3 Better gradients at fake samples implies stable GAN training

In GANs, the dynamics of the game depends in great part on $\nabla_{x_f} f(x_f)$ where x_f's are samples from the fake, or generated, distribution. This is because the generator only learns through the discriminator/critic and it uses $\nabla_{x_f} f(x_f)$ in order to improve its objective function. Thus, for stable training with a fixed or decreasing learning rate, $||\nabla_{x_f} f(x_f)||$ should not be too small.

The above means that, in order to get stable GAN training, we need to ensure that we obtain a solution with a stable non-zero gradient around fake samples. Thus, it is preferable to solve the penalized formulation from equation equation 6 and choose a large penalty term λ in order to obtain a small-gradient solution.

5.3 Are certain Margins better than others?

It is well known that L^p-norms (with $p \geq 1$) are more sensitive to outliers as p increases which is why many robust methods minimize the L^1-norm (Bloomfield & Steiger, 1983). Furthermore, minimizing the L^1-norm loss results in a median estimator (Bloomfield & Steiger, 1983). This suggests that penalizing the L^2 gradient norm penalty ($p = 2$) may not lead to the most robust classifier. We hypothesize that L^∞ gradient norm penalties may improve robustness in comparison to L^2 gradient norm penalties since they correspond to maximizing L^1-norm margin. In Section 6, we provide experimental evidence in support of our hypothesis.

6 Experiments

Following our analysis and discussion in the previous sections, we hypothesized that L^1 margins, corresponding to a L^∞ gradient norm penalty, would perform better than L^2 margins (L^2 gradient norm penalty). As far as we know, researchers have not yet tried using a L^∞ gradient norm penalty in

GANs. In addition, we showed that it would be more sensible to penalize violations of $||\nabla f(x)||_q \leq 1$ rather than $||\nabla f(x)||_q \approx 1$.

To test these hypotheses, we ran experiments on CIFAR-10 (a dataset of 60k images from 10 categories) (Krizhevsky et al., 2009) using HingeGAN ($L(z) = \max(0, 1 - z)$) and WGAN ($L(z) = z$) loss functions with L^1, L^2, L^∞ gradient norm penalties. We enforce either $||\nabla f(x)||_q \approx 1$ using Least Squares (LS) ($g(z) = (z-1)^2$) or $||\nabla f(x)||_q \leq 1$ using Hinge ($g(z) = \max(0, z-1)$). We used the standard hyperparameters: a learning rate (lr) of .0002, a batch size of 32, and the ADAM optimizer (Kingma & Ba, 2014) with parameters $(\alpha_1, \alpha_2) = (.50, .999)$ We used a DCGAN architecture (Radford et al., 2015). As per convention, we report the Fréchet Inception Distance (FID) (Heusel et al., 2017); lower values correspond to better generated outputs (higher quality and diversity). As per convention, all 50k images from the training part of the dataset were used for training and to calculate the FID. We ran all experiments using seed 1 and with gradient penalty $\lambda = 20$. Details on the architectures are in the Appendix. All models were trained using a single GPU. Code is available on *xxxxx*. The results are shown in Table 2.

Table 2: Fréchet Inception Distance (FID) after 100k generator iterations on CIFAR-10.

$g(\|\|\nabla_x f(x))\|\|_q)$	WGAN	HingeGAN
$(\|\|\nabla_x f(x))\|\|_1 - 1)^2$	99.7	88.9
$\max(0, \|\|\nabla_x f(x))\|\|_1 - 1)$	65.6	77.3
$(\|\|\nabla_x f(x))\|\|_2 - 1)^2$	37.6	32.8
$\max(0, \|\|\nabla_x f(x))\|\|_2 - 1)$	37.8	33.9
$(\|\|\nabla_x f(x))\|\|_\infty - 1)^2$	33.4	33.6
$\max(0, \|\|\nabla_x f(x))\|\|_\infty - 1)$	36	**27.1**

Due to space constraint, we only show the previously stated experiments in Table 2. However, we also ran additional experiments on CIFAR-10 with 1) Relativistic paired and average HingeGAN, 2) $\beta = (0, .90)$, 3) the standard CNN architecture from Miyato et al. (2018). Furthermore, we ran experiments on CAT (Zhang et al., 2008) with 1) Standard CNN (in 32x32), and 2) DCGAN (in 64x64). These experiments correspond to Table 3, 4, 5, 6, and 7 from the appendix.

In all sets of experiments, we generally observed that we obtain smaller FIDs by using: i) a larger q (as theorized), ii) the Hinge penalty to enforce an inequality gradient norm constraint (in both WGAN and HingeGAN), and iii) HingeGAN instead of WGAN.

7 Conclusion

This work provides a framework in which to derive MMCs that results in very effective GAN loss functions. In the future, this could be used to derive new gradient norm penalties which further improve the performance of GANs. Rather than trying to devise better ways of enforcing 1-Lipschitz, researchers may instead want to focus on constructing better MMCs (possibly by devising better margins).

This research shows a strong link between GANs with gradient penalties, Wasserstein's distance, and SVMs. Maximizing the minimum L^2-norm geometric margin, as done in SVMs, has been shown to lower bounds on the VC dimension which implies lower generalization error (Vapnik & Vapnik, 1998; Mount, 2015). This paper may help researchers bridge the gap needed to derive PAC bounds on Wasserstein's distance and GANs/IPMs with gradient penalty. Furthermore, it may be of interest to theoreticians whether certain margins lead to lower bounds on the VC dimension.

References

Adler, J. and Lunz, S. Banach wasserstein gan. In *Advances in Neural Information Processing Systems*, pp. 6754–6763, 2018.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223, 2017.

Bloomfield, P. and Steiger, W. L. *Least absolute deviations: theory, applications, and algorithms*. Springer, 1983.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Elsayed, G., Krishnan, D., Mobahi, H., Regan, K., and Bengio, S. Large margin deep networks for classification. In *Advances in neural information processing systems*, pp. 842–852, 2018.

Fedus, W., Rosca, M., Lakshminarayanan, B., Dai, A. M., Mohamed, S., and Goodfellow, I. Many paths to equilibrium: Gans do not need to decrease adivergence at every step. *arXiv preprint arXiv:1710.08446*, 2017a.

Fedus, W., Rosca, M., Lakshminarayanan, B., Dai, A. M., Mohamed, S., and Goodfellow, I. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. *arXiv preprint arXiv:1710.08446*, 2017b.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf.

Hestenes, M. R. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4 (5):303–320, 1969.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. 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, 2017.

Hölder, O. via an averaging clause. *Messages from the Society of Sciences and the Georg-Augusts-Universität zu Göttingen*, 1889:38–47, 1889.

Jolicoeur-Martineau, A. Gans beyond divergence minimization. *arXiv preprint arXiv:xxxx*, 2018a.

Jolicoeur-Martineau, A. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018b.

Jolicoeur-Martineau, A. On relativistic f-divergences. *arXiv preprint arXiv:1901.02474*, 2019.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Karush, W. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*, 1939.

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

Kodali, N., Abernethy, J., Hays, J., and Kira, Z. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Kuhn, H. W. and Tucker, A. W. Nonlinear programming, in (j. neyman, ed.) proceedings of the second berkeley symposium on mathematical statistics and probability, 1951.

Lim, J. H. and Ye, J. C. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2813–2821. IEEE, 2017.

Marsland, S. *Machine learning: an algorithmic perspective*. CRC press, 2015.

Matyasko, A. and Chau, L.-P. Margin maximization for robust classification using deep learning. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 300–307. IEEE, 2017.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for gans do actually converge? *arXiv preprint arXiv:1801.04406*, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Mount, J. How sure are you that large margin implies low vc dimension? *Win-Vector Blog*, 2015.

Müller, A. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

Petzka, H., Fischer, A., and Lukovnicov, D. On the regularization of wasserstein gans. *arXiv preprint arXiv:1709.08894*, 2017.

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

Rogers, L. J. An extension of a certain theorem in inequalities. *Messenger of Math.*, 17:145–150, 1888.

Rudin, W. Functional analysis, mcgrawhill. *Inc, New York*, 1991.

Vapnik, V. and Vapnik, V. Statistical learning theory. *Wiley*, 1998.

Zhang, W., Sun, J., and Tang, X. Cat head detection-how to effectively exploit shape and texture features. In *European Conference on Computer Vision*, pp. 802–816. Springer, 2008.

APPENDICES

A DETAILS ON EXPERIMENTS FOR TABLE 1

The classifier was a 4 layers fully-connected neural network with ReLU activation functions. It was trained for $10K$ iterations with the cross-entropy loss, the ADAM optimizer, a batch size of 256, $\lambda = 10$, and a learning rate of $.00005$. The margins (as in equation 3) were estimated using Gradient Descent with the Augmented Lagrangian method (Hestenes, 1969) until $|f(x_0)| < .01$. We estimated the expected margin using the average margin from 256 random samples of both classes.

B ADDITIONAL EXPERIMENTS

Note that the *smooth maximum* is defined as

$$\text{smax}(x_{(1)}, \dots, x_{(k)}) = \frac{\sum_{i=1}^{k} x_{(i)} e^{x_i}}{\sum_{i=1}^{k} e^{x_i}}.$$

We sometime use the smooth maximum as a smooth alternative to the L^∞-norm margin; results are worse with it.

Table 3: FID after 100k generator iterations on CIFAR-10 using the same setting as Table1, but we are using Relativistic paired and average GANs.

$g(\|\|\nabla_x f(x))\|\|_q)$	RpHinge	RaHinge
$(\|\|\nabla_x f(x))\|\|_1 - 1)^2$	64.4	65.0
$\max(0, \|\|\nabla_x f(x))\|\|_1 - 1)$	60.3	68.5
$(\|\|\nabla_x f(x))\|\|_2 - 1)^2$	32.8	31.9
$\max(0, \|\|\nabla_x f(x))\|\|_2 - 1)$	32.6	35.0
$(\|\|\nabla_x f(x))\|\|_\infty - 1)^2$	32.5	33.5
$\max(0, \|\|\nabla_x f(x))\|\|_\infty - 1)$	**28.2**	28.4
$(\text{smax}\|\nabla_x f(x)\| - 1)^2$	133.7	124.0
$\max(0, \text{smax}\|\nabla_x f(x)\| - 1)$	30.5	30.3

Table 4: FID after 100k generator iterations on CIFAR-10 using the same setting as Table1, but we are using Adam $\beta = (0, .90)$.

$g(\|\|\nabla_x f(x))\|\|_q)$	WGAN	HingeGAN
$(\|\|\nabla_x f(x))\|\|_1 - 1)^2$	163.2	179.0
$\max(0, \|\|\nabla_x f(x))\|\|_1 - 1)$	66.9	66.1
$(\|\|\nabla_x f(x))\|\|_2 - 1)^2$	34.6	33.8
$\max(0, \|\|\nabla_x f(x))\|\|_2 - 1)$	37.0	34.9
$(\|\|\nabla_x f(x))\|\|_\infty - 1)^2$	37.5	33.9
$\max(0, \|\|\nabla_x f(x))\|\|_\infty - 1)$	38.3	**28.6**
$(\text{smax}\|\nabla_x f(x)\| - 1)^2$	31.9	283.1
$\max(0, \text{smax}\|\nabla_x f(x)\| - 1)$	31.8	32.1

C MARGINS IN RELATIVISTIC GANS

Relativistic paired GANs (RpGANs) and Relativistic average GANs (RaGAN) (Jolicoeur-Martineau, 2018b; 2019) are GAN variants which tend to be more stable than their non-relativistic counterparts. These methods are not yet well understood and its unclear how to incorporate gradient penalty from our framework into these approach. In this subsection, we explain how we can link both approaches to MMCs.

Table 5: FID after 100k generator iterations on CIFAR-10 using the same setting as Table1, but we are using the standard CNN architecture.

$g(\|\|\nabla_x f(x))\|\|_q)$	WGAN	HingeGAN
$(\|\|\nabla_x f(x))\|\|_2 - 1)^2$	30.2	26.1
$\max(0, \|\|\nabla_x f(x))\|\|_2 - 1)$	31.8	27.3
$\max(0, \|\|\nabla_x f(x))\|\|_\infty - 1)$	74.3	**21.3**

Table 6: FID after 100k generator iterations on CAT (in 32x32) using the same setting as Table 1, but we are using the standard CNN architecture. Exceptionally, this set of experiment showed convergence at around 10-40k iterations (this has not been the case in any of the other experiments). For this reason, we show also the lowest FID obtained during training (FID was measured at every 10k iterations).

$g(\|\|\nabla_x f(x))\|\|_q)$	WGAN	HingeGAN
At 100k iterations		
$(\|\|\nabla_x f(x))\|\|_2 - 1)^2$	27.3	**19.5**
$\max(0, \|\|\nabla_x f(x))\|\|_2 - 1)$	21.4	23.9
$\max(0, \|\|\nabla_x f(x))\|\|_\infty - 1)$	66.5	24.0
Lowest FID obtained		
$(\|\|\nabla_x f(x))\|\|_2 - 1)^2$	20.9	16.2
$\max(0, \|\|\nabla_x f(x))\|\|_2 - 1)$	19.4	17.0
$\max(0, \|\|\nabla_x f(x))\|\|_\infty - 1)$	32.32	**9.5**

Table 7: FID after 100k generator iterations on CAT (in 64x64) using the same setting as Table 1.

$g(\|\|\nabla_x f(x))\|\|_q)$	WGAN	HingeGAN
$(\|\|\nabla_x f(x))\|\|_2 - 1)^2$	48.2	26.7
$\max(0, \|\|\nabla_x f(x))\|\|_2 - 1)$	43.7	29.6
$\max(0, \|\|\nabla_x f(x))\|\|_\infty - 1)$	18.3	**17.5**

Table 8: Extras from Table 1

$g(\|\|\nabla_x f(x))\|\|_q)$	WGAN	HingeGAN
$(\text{smax}\|\nabla_x f(x)\| - 1)^2$	35.3	197.9
$\max(0, \text{smax}\|\nabla_x f(x)\| - 1)$	31.4	29.5

Relativistic paired GANs (RpGANs) are defined as:

$$\max_{C:\mathcal{X}\to\mathbb{R}} \mathop{\mathbb{E}}_{\substack{x_1\sim\mathbb{P}\\ z\sim\mathbb{Z}}} \left[f\left(C(x_1) - C(G(z))\right)\right],$$
$$\max_{G} \mathop{\mathbb{E}}_{\substack{x_1\sim\mathbb{P}\\ z\sim\mathbb{Z}}} \left[f\left(C(G(z)) - C(x_1)\right)\right],$$

and Relativistic average GANs (RaGANs) are defined as:

$$\max_{C:\mathcal{X}\to\mathbb{R}} \mathbb{E}_{x_1\sim\mathbb{P}}\left[f_1\left(C(x_1) - \mathbb{E}_{z\sim\mathbb{Z}} C(G(z))\right)\right] + \mathbb{E}_{z\sim\mathbb{Z}}\left[f_2\left(C(G(z)) - \mathbb{E}_{x_1\sim\mathbb{P}} C(x_1)\right)\right],$$
$$\max_{G} \mathbb{E}_{z\sim\mathbb{Z}}\left[f_1\left(C(G(z)) - \mathbb{E}_{x_1\sim\mathbb{P}} C(x_1)\right)\right] + \mathbb{E}_{x_1\sim\mathbb{P}}\left[f_2\left(C(x_1) - \mathbb{E}_{z\sim\mathbb{Z}} C(G(z))\right)\right],$$

where $f, f_1, f_2 : \mathbb{R} \to \mathbb{R}$.

Most loss functions can be represented as RaGANs or RpGANs; SGAN, LSGAN, and HingeGAN all have relativistic counterparts.

C.1 RELATIVISTIC AVERAGE GANS

From the loss function of RaGAN, we can deduce its decision boundary. Contrary to typical classifiers, we define two boundaries, depending on the label. The two surfaces are defined as two sets of points (x_0, y_0) such that:

$$f(x_0) = \mathbb{E}_{x \sim \mathbb{Q}}[f(x)], \text{ when } y_0 = 1 (\text{real})$$
$$f(x_0) = \mathbb{E}_{x \sim \mathbb{P}}[f(x)], \text{ when } y_0 = -1 (\text{fake})$$

It can be shown that the relativistic average geometric margin is approximated as:

$$\begin{aligned}\gamma_p^{Ra}(x, y) \approx & \frac{((y+1)/2)(f(x) - \mathbb{E}_{x \sim \mathbb{Q}}[f(x)])}{||\nabla_x f(x)||_q} + \\ & \frac{((y-1)/2)(f(x) - \mathbb{E}_{x \sim \mathbb{P}}[f(x)])}{||\nabla_x f(x)||_q} \\ = & \frac{\alpha_{Ra}(x, y)}{\beta(x)}.\end{aligned}$$

Maximizing the boundary of RaGANs can be done in the following way:

$$\min_f \mathbb{E}_{(x,y) \sim \mathbb{D}} \left[L(\alpha_{Ra}(x, y)) + \lambda g(||\nabla_x f(x)||_q) \right].$$

C.2 RELATIVISTIC PAIRED GANS

From its loss function (as described in section 2.4), it is not clear what the boundary of RpGANs can be. However, through reverse engineering, it is possible to realize that the boundary is the same as the one from non-relativistic GANs, but using a different margin. We previously derived that the approximated margin (non-geometric) for any point is $\gamma_p(x) \approx \frac{|f(x)|}{||\nabla_x f(x)||_q}$. We define the geometric margin as the margin after replacing $|f(x)|$ by $yf(x)$ so that it depends on both x and y. However, there is an alternative way to transform the margin in order to achieve a classifier. We call it the *relativistic paired margin*:

$$\begin{aligned}\gamma_p^*(x_1, x_2) &= \gamma_p(x_1) - \gamma_p(x_2) \\ &\approx \frac{f(x_1)}{||\nabla_{x_1} f(x_1)||_q} - \frac{f(x_2)}{||\nabla_{x_2} f(x_2)||_q}.\end{aligned}$$

where x_1 is a sample from $\mathbb{P}$ and x_2 is a sample from $\mathbb{Q}$. This alternate margin does not depend on the label y, but only ask that for any pair of class 1 (real) and class 2 (fake) samples, we maximize the relativistic paired margin. This margin is hard to work with, but if we enforce $||\nabla_{x_1} f(x_1)||_q \approx ||\nabla_{x_2} f(x_2)||_q$, for all $x_1 \sim \mathbb{P}$,$x_2 \sim \mathbb{Q}$, we have that:

$$\gamma_p^*(x_1, x_2) \approx \frac{f(x_1) - f(x_2)}{||\nabla_x f(x)||_q},$$

where x is any sample (from class 1 or 2).

Thus, we can train an MMC to maximize the relativistic paired margin in the following way:

$$\begin{aligned}\min_f \mathop{\mathbb{E}}_{\substack{x_1 \sim \mathbb{P} \\ z \sim \mathbb{Z}}} & [L(f(x_1) - f(G(z)))] + \\ & \lambda \mathbb{E}_{(x,y) \sim \mathbb{D}} \left[g(||\nabla_x f(x)||_q) \right],\end{aligned}$$

where g must constrains $||\nabla_x f(x)||_q$ to a constant.

This means that minimizing $L(f(x_1) - f(x_2))$ without gradient penalty can be problematic if we have different gradient norms at samples from class 1 (real) and 2 (fake). This provides an explanation as to why RpGANs do not perform very well unless using a gradient penalty (Jolicoeur-Martineau, 2018b).

D PROOFS

Note that both of the following formulations represent the margin:

$$\begin{aligned}\gamma(x) &= \min_{x_0} ||x_0 - x|| \quad \text{s.t.} \quad f(x_0) = 0 \\ &= \min_{r} ||r|| \quad \text{s.t.} \quad f(x + r) = 0\end{aligned}$$

D.1 BOUNDED GRADIENT $\iff$ LIPSCHITZ

Assume that $f : X \to \mathbb{R}$ and X is a convex set.

Let $\tilde{x}(\alpha) = \alpha x_1 + (1 - \alpha)x_2$, where $\alpha \in [0, 1]$ be the interpolation between any two points $x_1, x_2 \in X$. We know that $\tilde{x}(\alpha) \in X$ for any $\alpha \in [0, 1]$ by convexity of X.

$$\begin{aligned}f(x_1) - f(x_2) &= f(\tilde{x}(1)) - f(\tilde{x}(0)) \\ &= \int_0^1 \frac{df(\tilde{x}(\alpha))}{d\alpha} d\alpha \\ &= \int_0^1 \nabla f(\tilde{x}(\alpha)) \frac{\tilde{x}(\alpha)}{d\alpha} d\alpha \\ &= \int_0^1 \nabla f(\tilde{x}(\alpha))(x_1 - x_2) d\alpha \\ &= (x_1 - x_2) \int_0^1 \nabla f(\tilde{x}(\alpha)) d\alpha\end{aligned}$$

1) We show $||\nabla f(x)||_p \leq K \implies \frac{|f(x)-f(y)|}{||x-y||_p} \leq K$ for all x, y:

Let $||\nabla f(x)||_p \leq K$ for all $x \in X$.

$$\begin{aligned}|f(x_1) - f(x_2)| &= ||(x_1 - x_2) \int_0^1 \nabla f(\tilde{x}(\alpha)) d\alpha||_p \\ &\leq ||x_1 - x_2||_p \cdot ||_p \int_0^1 \nabla f(\tilde{x}(\alpha)) d\alpha|| \\ &\leq ||x_1 - x_2||_p \int_0^1 ||\nabla f(\tilde{x}(\alpha))||_p d\alpha \\ &\leq ||x_1 - x_2||_p \int_0^1 K d\alpha \\ &\leq K||x_1 - x_2||_p\end{aligned}$$

2) We show $\frac{|f(x)-f(y)|}{||x-y||_p} \leq K$ for all $x, y \implies ||\nabla f(x)||_p \leq K$ for $p \geq 2$:

Assume $p \geq 2$ and $\frac{1}{p} + \frac{1}{q} = 1$.

$$
\begin{aligned}
||\nabla_x f(x)||_p &\leq ||\nabla_x f(x)||_q \quad \text{since } p \geq q \\
&= \max_v \nabla_x f(x)^T v \quad \text{s.t.} \quad ||v||_p \leq 1 \\
&= \left|\nabla_x f(x)^T v^*\right| \quad \text{where } v^* \text{ is the optimum} \\
&= lim_{h \to 0} \left| \frac{f(x + hv^*) - f(x)}{h} \right| \\
&\leq lim_{h \to 0} \frac{|f(x + hv^*) - f(x)|}{h||v^*||_p} \\
&= lim_{h \to 0} \frac{|f(x + hv^*) - f(x)|}{||x + hv^* - x||_p} \\
&\leq K
\end{aligned}
$$

D.2 TAYLOR APPROXIMATION

Let $r = x_0 - x$; at the boundary x_0, we have that $f(x_0) = f(x+r) = C$, for some constant C. In the paper, we use generally assume $C = 0$. We will make use of the following Taylor approximations:

$$
\begin{aligned}
&f(x + r) \approx f(x) + \nabla_x f(x)^T r \\
\implies &f(x) - C \approx -\nabla_x f(x)^T r
\end{aligned}
$$

and

$$
\begin{aligned}
&f(x) \approx f(x_0) + \nabla_{x_0} f(x_0)^T (x - x_0) \\
\implies &f(x) - C \approx -\nabla_{x_0} f(x_0)^T r
\end{aligned}
$$

D.3 TAYLOR APPROXIMATION (AFTER SOLVING)

To make things easier, we maximize $(\gamma_2(x))^2$ instead of $\gamma_2(x)$. We also maximize with respect to x_0 instead of r:

$$
\begin{aligned}
\gamma_2^2(x) &= \min_{x_0} ||x_0 - x||_2^2 \quad \text{s.t.} \quad f(x_0) = 0 \\
&= \min_{x_0} ||x_0 - x||_2^2 - \lambda f(x_0),
\end{aligned}
$$

where λ is a scalar (Lagrange multiplier). We can then differentiate with respect to x_0:

$$
\nabla_{x_0} \gamma_2^2(x) = (x_0 - x) + \lambda \nabla_{x_0} f(x_0) = 0. \tag{7}
$$

We will then use a inner product to be able to extract the optimal Lagrange multiplier:

$$
\begin{aligned}
&\implies -\nabla_{x_0} f(x_0)^T (x_0 - x) = \lambda^* \nabla_{x_0} f(x_0)^T \nabla_{x_0} f(x_0) \\
&\implies \lambda^* = -\frac{\nabla_{x_0} f(x_0)^T (x_0 - x)}{\nabla_{x_0} f(x_0)^T \nabla_{x_0} f(x_0)} \\
&\implies \lambda^* = -\frac{\nabla_{x_0} f(x_0)^T (x_0 - x)}{||\nabla_{x_0} f(x_0)||_2^2}
\end{aligned}
$$

Now, we plug-in the optimal Langrange multiplier into equation equation 7 and we use a inner product:

$$
\begin{aligned}
&\implies x_0 - x = -\lambda \nabla_{x_0} f(x_0) \\
&\implies x_0 - x = \frac{\nabla_{x_0} f(x_0)^T (x_0 - x)}{||\nabla_{x_0} f(x_0)||_2^2} \nabla_{x_0} f(x_0) \\
&\implies (x_0 - x)^T (x_0 - x) = \frac{\left(\nabla_{x_0} f(x_0)^T (x_0 - x)\right)^2}{||\nabla_{x_0} f(x_0)||_2^2} \\
&\implies \gamma_2^2(x) = ||x_0 - x||_2^2 = \frac{\left(\nabla_{x_0} f(x_0)^T (x_0 - x)\right)^2}{||\nabla_{x_0} f(x_0)||_2^2} \\
&\implies \gamma_2(x) = \frac{\left|\nabla_{x_0} f(x_0)^T (x_0 - x)\right|}{||\nabla_{x_0} f(x_0)||_2}
\end{aligned}
$$

D.4 TAYLOR APPROXIMATION (BEFORE SOLVING)

$$
\begin{aligned}
&\min_r ||r||_p \quad \text{s.t.} \quad f(x+r) = C \\
&\approx \min_r ||r||_p \quad \text{s.t.} \quad f(x) + \nabla_x f(x)^T r = C \\
\implies &\min_r ||r||_p = \frac{|f(x) - C|}{\max_r \frac{|\nabla_x f(x)^T r|}{||r||_p}} \\
&= \frac{|f(x) - C|}{\max_r \frac{\nabla_x f(x)^T r}{||r||_p}} \\
&= \frac{|f(x) - C|}{\max_r \nabla_x f(x)^T \frac{r}{||r||_p}} \\
&= \frac{|f(x) - C|}{\max_{||r||_p \leq 1} \nabla_x f(x)^T r} \\
&= \frac{|f(x) - C|}{||\nabla_x f(x)||_q},
\end{aligned}
$$

where $\frac{1}{p} + \frac{1}{q} = 1$ This is true because of the definition of the Dual norm (Rudin, 1991):

$$
||a||_* = \max_{||r||_p \leq 1} a^T r = \max_r \frac{a^T r}{||r||_p} = ||a||_q
$$

For a standard classifier, we have $C = 0$. For a RaGAN, we have $C = \mathbb{E}_{\mathbb{Q}}[f(x)]$ when $y = 1$ (real) and $C = \mathbb{E}_{\mathbb{P}}[f(x)]$ when $y = -1$ (fake).

E ARCHITECTURES

E.1 DCGAN 32X32 (AS IN TABLE 1, 2, 3)

Generator
$z \in \mathbb{R}^{128} \sim N(0, I)$
ConvTranspose2d 4x4, stride 1, pad 0, 128→512
BN and ReLU
ConvTranspose2d 4x4, stride 2, pad 1, 512→256
BN and ReLU
ConvTranspose2d 4x4, stride 2, pad 1, 256→128
BN and ReLU
ConvTranspose2d 4x4, stride 2, pad 1, 128→3
Tanh

Discriminator
$x \in \mathbb{R}^{3\text{x}32\text{x}32}$
Conv2d 4x4, stride 2, pad 1, 3→128
LeakyReLU 0.2
Conv2d 4x4, stride 2, pad 1, 128→256
BN and LeakyReLU 0.2
Conv2d 4x4, stride 2, pad 1, 256→512
BN and LeakyReLU 0.2
Conv2d 4x4, stride 2, pad 1, 512→1

E.2 DCGAN 64X64 (AS IN TABLE 6)

Generator
$z \in \mathbb{R}^{128} \sim N(0, I)$
ConvTranspose2d 4x4, stride 1, pad 0, 128→512
BN and ReLU
ConvTranspose2d 4x4, stride 2, pad 1, 512→256
BN and ReLU
ConvTranspose2d 4x4, stride 2, pad 1, 256→128
BN and ReLU
ConvTranspose2d 4x4, stride 2, pad 1, 128→64
BN and ReLU
ConvTranspose2d 4x4, stride 2, pad 1, 64→3
Tanh

Discriminator
$x \in \mathbb{R}^{3\text{x}64\text{x}64}$
Conv2d 4x4, stride 2, pad 1, 3→64
LeakyReLU 0.2
Conv2d 4x4, stride 2, pad 1, 64→128
BN and LeakyReLU 0.2
Conv2d 4x4, stride 2, pad 1, 128→256
BN and LeakyReLU 0.2
Conv2d 4x4, stride 2, pad 1, 256→512
BN and LeakyReLU 0.2
Conv2d 4x4, stride 2, pad 1, 512→1

E.3 STANDARD CNN (AS IN TABLE 4, 5)

Generator
$z \in \mathbb{R}^{128} \sim N(0, I)$
linear, 128 → 512*4*4
Reshape, 512*4*4 → 512 x 4 x 4
ConvTranspose2d 4x4, stride 2, pad 1, 512→256
BN and ReLU
ConvTranspose2d 4x4, stride 2, pad 1, 256→128
BN and ReLU
ConvTranspose2d 4x4, stride 2, pad 1, 128→64
BN and ReLU
ConvTranspose2d 3x3, stride 1, pad 1, 64→3
Tanh

Discriminator
$x \in \mathbb{R}^{3\text{x}32\text{x}32}$
Conv2d 3x3, stride 1, pad 1, 3→64
LeakyReLU 0.1
Conv2d 4x4, stride 2, pad 1, 64→64
LeakyReLU 0.1
Conv2d 3x3, stride 1, pad 1, 64→128
LeakyReLU 0.1
Conv2d 4x4, stride 2, pad 1, 128→128
LeakyReLU 0.1
Conv2d 3x3, stride 1, pad 1, 128→256
LeakyReLU 0.1
Conv2d 4x4, stride 2, pad 1, 256→256
LeakyReLU 0.1
Conv2d 3x3, stride 1, pad 1, 256→512
Reshape, 512 x 4 x 4 → 512*4*4
linear, 512*4*4 → 1