GENERALIZATION AND STABILITY OF GANS: A THEORY AND PROMISE FROM DATA AUGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The instability when training generative adversarial networks (GANs) is a notoriously difficult issue, and the generalization of GANs remains open. In this paper, we will analyze various sources of instability which not only come from the discriminator but also the generator. We then point out that the requirement of Lipschitz continuity on both the discriminator and generator leads to generalization and stability for GANs. As a consequence, this work naturally provides a generalization bound for a large class of existing models and explains the success of recent large-scale generators. Finally, we show why data augmentation penalizes the gradients of both the discriminator and generator. This work therefore provides a theoretical basis for a simple way to ensure generalization in GANs, explaining the highly successful use of data augmentation for GANs in practice.

1 INTRODUCTION

In *Generative Adversarial Networks* (GAN) (Goodfellow et al., 2014), we often want to learn a discriminator D and a generator G by solving the following minimax problem:

$$\min_G \max_D \mathbb{E}_{x\sim p_d} \log(D(x)) + \mathbb{E}_{z\sim p_z} \log(1 - D(G(z))) \tag{1}$$

where p_d is a data distribution that generates real data, and p_z is some noise distribution. G can be parameterized by a neural network to map a noise z to a point in the data space. After training, G can be used to generate novel data which are as realistic as possible.

Since its introduction by Goodfellow et al. (2014), a significant progress has been made for developing GANs and for interesting applications (Hong et al., 2019). Some recent works (Brock et al., 2019; Zhang et al., 2019) can train a generator that produces synthesis images of extremely high quality. Nevertheless, little is known about the generalization of the trained players, and the training is nororiously challenging due to instability (Salimans et al., 2016; Arjovsky & Bottou, 2017; Gulrajani et al., 2017; Kodali et al., 2017; Fedus et al., 2018; Miyato et al., 2018; Kurach et al., 2019; Wu et al., 2019; Thanh-Tung et al., 2019; Xu et al., 2020; Chu et al., 2020).

This work has the following contributions:

- We make a bridge between Lipschitz continuity of a loss and generalization of a hypothesis. Basically, we show that the learnt hypothesis will generalize well if the training loss is Lipschitz continuous w.r.t *input*. This result enables us to easily analyze the players in a large class of GANs, and provides their generalization bounds. Those bounds theoretically explain the success of various existing techniques to stabilize GAN training, including gradient penalty (Gulrajani et al., 2017) and spectral normalization (Miyato et al., 2018), and the success of recent large-scale generators (Brock et al., 2019; Zhang et al., 2019).
- We analyze various sources of instability, including the discriminator, the generator, and the noise. We point out why existing understandings are insufficient to deal with instability.
- Finally, we show why data augmentation implicitly imposes a gradient penalty, and explain how to properly use this simple technique to ensure small Lipschitz constants of the discriminator and generator. As a result, we provide a theory for explaining the highly successful use of data augmentation for GANs in recent practices (Zhao et al., 2020c; Zhang et al., 2020; Zhao et al., 2020a).

2 RELATED WORK

Generalization: There are few efforts to analyze the generalization for GANs using the notion of *neural distance*, $d_{\mathcal{D}}(\mu, \nu)$ which is the distance of two distributions (μ, ν). Arora et al. (2017) show that we can bound the quantity $|d_{\mathcal{D}}(\mu, \nu) - d_{\mathcal{D}}(\mu_m, \nu_m)|$, where (μ_m, ν_m) are empirical versions of (μ, ν). Zhang et al. (2018) and Jiang et al. (2019) analyze $|d_{\mathcal{D}}(\mu, \nu_m) - \inf_\nu d_{\mathcal{D}}(\mu, \nu)|$ to see generalization of G. To obtain those bounds, we need the assumption of Lipschitz continuity of D w.r.t its parameters. Note that those notions of generalization are different from the classical sense, because the neural distance is defined on the *best* discriminator in its family $\mathcal{D}$, suggesting that the distance can be zero even when μ and ν are far away. Indeed, Arjovsky & Bottou (2017) show that there exists a perfect but constant discriminator whenever the supports of μ and ν are non-overlapping. Therefore, existing bounds based on neural distance are insufficient to see the generalization of the learnt hypothesis. On the other hand, Qi (2020) shows a generalization bound of the generator for their proposed Loss-Sensitive GAN. Nonetheless, it is nontrivial to make their bound to work with other GAN losses. Wu et al. (2019) show that the discriminator will generalize if the learning algorithm is differentially private. Their concept of *differential privacy* basically requires that the learnt hypothesis will change negligibly if the training set slightly changes. Such a requirement is known as *algorithmic stability* (Xu et al., 2010) and is nontrivial to assure in practice. In contrast, we show that under the assumption of Lipschitz continuity of the loss w.r.t input, both D and G generalize well in the classical sense. Our bounds apply to a large class of GANs.

Techniques for stabilizing GAN training: Instability can come from different sources, such as gradient vanishing (Arjovsky & Bottou, 2017), gradient exploding (Gulrajani et al., 2017; Thanh-Tung et al., 2019), gradient uninformativeness (Zhou et al., 2019), ill-condition of the Jacobian of the players w.r.t their parameters (Mescheder et al., 2017; Nie & Patel, 2019). Since instability appears frequently and really challenging, a lot of approaches have been proposed and can be divided into four main groups: *network architecture* (Radford et al., 2016; Salimans et al., 2016; Zhang et al., 2019; Karras et al., 2018; 2019; Karnewar & Wang, 2020; Schonfeld et al., 2020), *training loss* (Mao et al., 2017; 2019; Zhao et al., 2017; Arjovsky et al., 2017; Li et al., 2017; Jolicoeur-Martineau, 2019; Qi, 2020; Guo et al., 2020), *optimization method* (Heusel et al., 2017; Yazıcı et al., 2019; Wang et al., 2020; Chavdarova et al., 2020; Chu et al., 2020), and *regularization* (Gulrajani et al., 2017; Mescheder et al., 2017; Nagarajan & Kolter, 2017; Roth et al., 2017; Sanjabi et al., 2018; Miyato et al., 2018; Guo et al., 2019; Nie & Patel, 2019; Zhou et al., 2019; Qi, 2020; Zhang et al., 2020; Xu et al., 2020). Our work identifies further sources of instability, and theoretically shows why data augmentation (a very cheap and practical way) can help stabilizing GAN training.

What's missing in the existing literature about the connection between Lipschitz continuity, stability, and generalization? Lipschitz continuity naturally appears in the formulation of Wasserstein GAN (WGAN) (Arjovsky et al., 2017). It was then quickly recognized as a crucial component to improve various GANs (Fedus et al., 2018; Lucic et al., 2018; Mescheder et al., 2018; Kurach et al., 2019; Jenni & Favaro, 2019; Wu et al., 2019; Zhou et al., 2019; Qi, 2020; Chu et al., 2020). Gradient penalty and spectral normalization are two popular techniques to constraint the Lipschitz continuity of D or G w.r.t their *inputs*. Some other works (Mescheder et al., 2017; Nagarajan & Kolter, 2017; Sanjabi et al., 2018; Nie & Patel, 2019) suggest to control the Lipschitz continuity of D or G w.r.t their *parameters*. From extensive experiments, those works found that Lipschitz continuity can help improving stability and quality of GANs. However, we will point out why Lipschitz constraints only on D or G are insufficient to ensure generalization and stability.

Continuity in parameters versus inputs: It is worth mentioning that there are the two kinds of Lipschitz constraint: continuity in inputs (Gulrajani et al., 2017; Jenni & Favaro, 2019; Wu et al., 2019; Zhou et al., 2019; Qi, 2020), and continuity in parameters (Mescheder et al., 2017; Nagarajan & Kolter, 2017; Sanjabi et al., 2018; Nie & Patel, 2019). Note that parameter continuity does not always imply input continuity. Chu et al. (2020) suggests that both D and G need to be smooth in both their inputs and parameters in order to ensure that the learning for generator will converge, given a fixed discriminator. It is worth noting that optimization convergence does not always imply good generalization for GANs. Our work shows that Lipschitz continuity of the training loss w.r.t. input is sufficient to guarantee generalization for both D and G.

Data augmentation for GANs: Data augmentation (Shorten & Khoshgoftaar, 2019) has been playing a crucial role in various areas. Some recent works (Zhao et al., 2020c; Zhang et al., 2020; Zhao et al.,

2020a) found that it is really beneficial to exploit data augmentation for training GANs. However, there is a lack of theory to explain those observations. This work will fill this gap.

3 LIPSCHITZ CONTINUITY, ROBUSTNESS, AND GENERALIZATION

In this section, we will review an important result by Xu & Mannor (2012) about the close relation between the robustness and generalization of a learning algorithm. We then show how robustness connects to Lipschitz continuity. Finally we point out why imposing a Lipschitz constraint on the loss will lead to generalization and discuss an application to GANs.

Consider a *learning problem* specified by a hypothesis class $\mathbb{H}$, an instance set $\mathbb{Z}$, and a loss function $f : \mathbb{H} \times \mathbb{Z} \to \mathbb{R}$ which is bounded by a constant C. Given a distribution with density $p(z)$ defined on $\mathbb{Z}$, the quality of a hypothesis is measured by its *expected loss* $F(h) = \mathbb{E}_{z\sim p(z)}[f(h; z)]$. Since $p(z)$ is unknown, we need to rely on a finite training sample $\boldsymbol{S} = \{z_1, ..., z_m\} \subset \mathbb{Z}$ and often work with the empirical loss $F_S(h) = \frac{1}{m}\sum_{z\in\boldsymbol{S}} f(h; z)$. A *learning algorithm* $\mathcal{A}$ will pick a hypothesis based on input $\boldsymbol{S}$. One can interpret a learning algorithm as a mapping from a subset of $\mathbb{Z}$ to a hypothesis.

Let $\mathbb{Z} = \bigcup_{i=1}^{K} \mathbb{Z}_i$ be a partition of $\mathbb{Z}$ into K disjoint subsets. We use the following definition about robustness of an algorithm.

Definition 1 (Robustness) *An algorithm $\mathcal{A}$ is (K, ϵ)-**robust**, for $\epsilon(\cdot) : \mathbb{Z}^m \to \mathbb{R}$, if the following holds for all $\boldsymbol{S} \in \mathbb{Z}^m$:*

$$\forall s \in \boldsymbol{S}, \forall z \in \mathbb{Z}, \forall i \in \{1, ..., K\} : \text{ if } s, z \in \mathbb{Z}_i \text{ then } |f(\mathcal{A}(\boldsymbol{S}), s) - f(\mathcal{A}(\boldsymbol{S}), z)| \leq \epsilon(\boldsymbol{S}). \quad (2)$$

Basically, a robust algorithm will learn a hypothesis which ensures that the losses of two similar data instances should be the same. A small change in the input leads to a small change in the ouput of the learnt hypothesis. In other words, the robustness ensures that each testing sample which is close to the training dataset will have a similar loss with that of the closest training samples. Therefore, the hypothesis $\mathcal{A}(\boldsymbol{S})$ will generalize well over the areas around $\boldsymbol{S}$.

Theorem 1 (Xu & Mannor (2012)) *If a learning algorithm $\mathcal{A}$ is (K, ϵ)-robust, and the training data $\boldsymbol{S}$ is an i.i.d. sample from distribution $p(z)$, then for any $\delta \in (0, 1]$ we have the following with probability at least $1 - \delta$: $|F(\mathcal{A}(\boldsymbol{S})) - F_S(\mathcal{A}(\boldsymbol{S}))| \leq \epsilon(\boldsymbol{S}) + C\sqrt{(K \log 4 - 2 \log \delta)/m}$.*

This theorem formally makes the important connection between robustness and generalization of an algorithm. Essentially, an algorithm will generalize if it is robust. One important implication of this result is that we should ensure the robustness of a learning algorithm in practice. However, it is nontrivial to do so.

Let us have a closer look at robustness. $\epsilon(\boldsymbol{S})$ in fact bounds the amount of change in the ouput with respect to a change in the input of the loss function given a fixed hypothesis. This observation suggests that robustness closely resembles the concept of Lipschitz continuity. Remember that a function $y : \mathbb{Z} \to \mathbb{Y}$ is said to be *L-Lipschitz continuous* if $d_y(y(z), y(z')) \leq L d_z(z, z')$ for any $z, z' \in \mathbb{Z}$, where d_z is a metric on $\mathbb{Z}$, d_y is a metric on $\mathbb{Y}$, and $L \geq 0$ is the Lipschitz constant. Therefore, we establish the following connection between robustness and Lipschitz continuity.

Lemma 2 *Given any constant $\lambda > 0$, consider a loss $f : \mathbb{H} \times \mathbb{Z} \to \mathbb{R}$, where $\mathbb{Z} \subset \mathbb{R}^n$ is compact, $B = \sup_{z,z'\in\mathbb{Z}} d_z(z, z') = \sup_{z,z'\in\mathbb{Z}} ||z - z'||_\infty, K = \lceil B^n \lambda^{-n} \rceil$. If $f(\mathcal{A}; z)$ is L-Lipschitz continuous w.r.t input z, then algorithm $\mathcal{A}$ is $(K, L\lambda)$-robust.*

Proof: It is easy to see that there exist $K = \lceil (B/\lambda)^n \rceil$ disjoint n-dimensional cubes, each with edge length of λ, satisfying that their union covers $\mathbb{Z}$ completely since $\mathbb{Z}$ is compact. Let $\mathbb{C}_k$ be one of those cubes, indexed by k, and $\mathbb{Z}_k = \mathbb{Z} \cap \mathbb{C}_k$. We can write $\mathbb{Z} = \bigcup_{k=1}^{K} \mathbb{Z}_k$.

Consider any $s, z \in \mathbb{Z}$. If both s and z belong to the same $\mathbb{Z}_k$ for some k, we have $|f(\mathcal{A}; s) - f(\mathcal{A}; z)| \leq L||z - z'||_\infty \leq L\lambda$ due to the Lipschitz continuity of f, completing the proof. □

Combining Theorem 1 and Lemma 2, we make the following connection between Lipschitz continuity and generalization.

Theorem 3 (Lipschitz continuity ⇒ Generalization) *If a loss $f(\mathcal{A}; z)$ is L-Lipschitz continuous w.r.t input z in a compact set $\mathbb{Z} \subset \mathbb{R}^n$, and the training data $\boldsymbol{S}$ is an i.i.d. sample from distribution $p(z)$, then for any constants $\delta \in (0, 1]$ and $\lambda \in (0, B]$ we have the followings with probability at least $1 - \delta$: $|F(\mathcal{A}(\boldsymbol{S})) - F_S(\mathcal{A}(\boldsymbol{S}))| \leq L\lambda + \frac{1}{\sqrt{m}}\sqrt{\lceil B^n \lambda^{-n} \rceil \log 4 - \log \delta^2}$, and*

$$|F(\mathcal{A}(\boldsymbol{S})) - F_S(\mathcal{A}(\boldsymbol{S}))| \quad \leq \quad LB + \frac{1}{\sqrt{m}}\sqrt{\log 4 - \log \delta^2}. \tag{3}$$

This theorem tells that Lipschitz continuity is the key to ensure an algorithm $\mathcal{A}$ (and the learnt hypothesis) to generalize. $\mathcal{A}$ generalizes better as the Lipschitz constant of the loss decreases. Note that there is a tradeoff between the Lipschitz constant and the expected loss $F(\mathcal{A}(\boldsymbol{S}))$ of the learnt hypothesis. A smaller L means that both f and hypothesis $h = \mathcal{A}(\boldsymbol{S})$ are getting simpler and flatter, due to $\frac{\partial f}{\partial z} = \frac{\partial f}{\partial h}\frac{\partial h}{\partial z}$, and hence may increase $F(h)$. In contrast, a decrease of $F(h)$ may require h to be more complex and hence may increase the Lipschitz constants of both f and h. It is worth noting that the Lipschitz constant of the loss also depends on how fast the loss changes w.r.t h.

Application to GANs: Consider $v_d(D, G, x, z) = \psi_1(D(x)) + \psi_2(1 - D(G(z)))$ and $v_g(D, G, z) = \psi_3(1 - D(G(z)))$ being the losses defined from a real example $x \sim p_d$, a noise $z \sim p_z$, a discriminator D, and a generator G. Different choices of the *measuring functions* (ψ_1, ψ_2, ψ_3) will lead to different GANs. For example, the vanilla GAN (Goodfellow et al., 2014) uses $\psi_1(x) = \psi_2(x) = \psi_3(x) = \log(x)$; WGAN (Arjovsky et al., 2017) uses $\psi_1(x) = \psi_2(x) = \psi_3(x) = x$; LSGAN (Mao et al., 2017; 2019) uses $\psi_1(x) = -(x+a)^2$, $\psi_2(x) = -(x+b)^2$, $\psi_3(x) = (x+c)^2$ for some constants a, b, c; EBGAN (Zhao et al., 2017) uses $\psi_1(x) = x$, $\psi_3(x) = 1 - x$ and $\psi_2(x) = \max(0, r - x)$ for some constant r. Then the expected loss and empirical loss defined on $\boldsymbol{S} = \{x_i, z_i\}_{i=1}^m$ are:

$$\begin{aligned}
&\text{D loss:} && V_d = \mathbb{E}_{x\sim p_d, z\sim p_z} v_d(D, G, x, z) = \mathbb{E}_{x\sim p_d}\psi_1(D(x)) + \mathbb{E}_{z\sim p_z}\psi_2(1 - D(G(z))) \\
&\text{G loss:} && V_g = \mathbb{E}_{z\sim p_z} v_g(D, G, z) = \mathbb{E}_{z\sim p_z}\psi_3(1 - D(G(z))) \\
&\text{Empirical losses:} && V_{d,S} = \frac{1}{m}\sum_{(x,z)\in\boldsymbol{S}} v_d(D, G, x, z); \quad V_{g,S} = \frac{1}{m}\sum_{z\in\boldsymbol{S}} v_g(D, G, z)
\end{aligned}$$

It is worth observing that the Lipschitz continuity of v_d and v_g depends on that of the measuring functions and the players. Therefore, the following results readily come from Theorem 3.

Corollary 1 (Generalization in GANs) *Assume the training data $\boldsymbol{S} = \{x_i, z_i\}_{i=1}^m$ consists of m i.i.d. samples from real distribution p_d defined on a compact set $\mathbb{Z}_x \subset \mathbb{R}^n$ and m i.i.d. samples from fake distribution p_z defined on a compact set $\mathbb{Z}_z \subset \mathbb{R}^{n_z}$, denote $B_x = \sup_{x,x'\in\mathbb{Z}_x} ||x - x'||_\infty$, $B_z = \sup_{z,z'\in\mathbb{Z}_z} ||z - z'||_\infty$. Assume further that (ψ_1, ψ_2, ψ_3) are L_ψ-Lipschitz continuous. Given any constant $\delta \in (0, 1]$, the following generalization bounds hold with probability at least $1 - \delta$:*

1. *$|V_d - V_{d,S}| \leq L_\psi L_d B_x + \frac{1}{\sqrt{m}}\sqrt{\log 4 - \log \delta^2}$, for any given G, provided that D is L_d-Lipschitz continuous w.r.t x.*
2. *$|V_g - V_{g,S}| \leq L_\psi L_d L_g B_z + \frac{1}{\sqrt{m}}\sqrt{\log 4 - \log \delta^2}$, for any given D, provided that D is L_d-Lipschitz continuous w.r.t x and G is L_g-Lipschitz continuous w.r.t z.*

For many models, such as WGAN, the measuring functions and D are Lipschitz continuous w.r.t their inputs. As a result, the generalization bound for D in Corollary 1 naturally holds. Note that the generator in WGAN, LSGAN, and EBGAN will be Lipschitz continuous w.r.t z, if we use some regularization methods such as gradient penalty or spectral normalization for both players. This provides a significant evidence to explain the success of some large-scale generators (Zhang et al., 2019; Brock et al., 2019), which use spectral normalization. Unfortunately, the loss of the vanilla GAN and many other variants are not guaranteed to be Lipschitz continuous due to the use of $\log$ function. Therefore, the generalization bound (3) may not be directly applied.

One natural suggestion from Theorem 3 and Corollary 1 for improving generalization in GANs is either taking more training data or decreasing the Lipschitz constant of the GAN loss. It is worth observing that a small Lipschitz constant of the loss not only requires that both discriminator and generator are Lipschitz continuous w.r.t their inputs, but also requires the Lipschitz continuity of the

loss w.r.t both players. Some existing losses, such as saturating and non-saturating GANs (Goodfellow et al., 2014), do not satisfy the latter requirement. Interestingly, most existing efforts focus on ensuring the Lipschitz continuity of the players in GANs, and leave the loss open. As pointed out in Section 4, constraining on either discriminator or generator only is not sufficient to ensure Lipschitz continuity of the loss and therefore the generalization of those players.

Comparison with other generalization bounds: There is a line of works (Arora et al., 2017; Zhang et al., 2018; Jiang et al., 2019) that analyze the generalization for GANs in terms of *neural distance*. Let (μ, ν) be two distributions and (μ_m, ν_m) be their empirical versions estimated from a sample $\boldsymbol{S}$ of size m. We define the neural distances: $d_{\mathcal{D}}(\mu, \nu) = \sup_{D \in \mathcal{D}} |\mathbb{E}_{x \sim p_d, z \sim p_z} v_d(D, G, x, z)| - 2\psi(\frac{1}{2})$ and $d_{\mathcal{D}}(\mu_m, \nu_m) = \sup_{D \in \mathcal{D}} |\frac{1}{m} \sum_{(x,z) \in \boldsymbol{S}} v_d(D, G, x, z)| - 2\psi(\frac{1}{2})$, where $\psi(\cdot) = \psi_1(\cdot) = \psi_2(\cdot)$, $\mathcal{D}$ is a family of discriminators, ν is the distribution induced by generator $G(z)$, $z \sim p_z$. Arora et al. (2017) analyze the generalization by upper bounding the quantity $|d_{\mathcal{D}}(\mu, \nu) - d_{\mathcal{D}}(\mu_m, \nu_m)|$, while (Zhang et al., 2018; Jiang et al., 2019) analyze $|d_{\mathcal{D}}(\mu, \nu_m) - \inf_\nu d_{\mathcal{D}}(\mu, \nu)|$ to see generalization of G, under the assumption of Lipschitz continuity of D w.r.t its parameters. We emphasize that those notions of generalization are different from the classical sense. Indeed, the neural distance $d_{\mathcal{D}}(\mu, \nu)$ is defined on the *best* discriminator, suggesting that the distance can be zero even when μ and ν are far away. Note that there will exist a constant but perfect D, meaning $d_{\mathcal{D}}(\mu, \nu) = 0$, whenever μ and ν do not have overlapping supports (Arjovsky & Bottou, 2017). Therefore, existing bounds based on neural distance do not tell much about the generalization of GANs. In contrast, our work provides generalization bounds in the classical sense for both players.

Qi (2020) shows a generalization bound of the generator for their Loss-Sensitive GAN, which contains a Lipschitz regularizer for the generator and the margin between the real and fake distributions. Nonetheless, it is nontrivial to make their bound to work with other GAN losses. Wu et al. (2019) show that the discriminator will generalize if the learning algorithm is differentially private. Their concept of *differential privacy* basically requires that the learnt hypothesis $h = \mathcal{A}(\boldsymbol{S})$ will change negligibly if the training sample $\boldsymbol{S}$ slightly changes. Such a requirement is known as *algorithmic stability* (Xu et al., 2010) and resembles the Lipschitz continuity of the learning algorithm $\mathcal{A}(\boldsymbol{S})$. It is worth observing that the Lipschitz continuity of an algorithm is a strong assumption and is nontrivial to assure in practice. In contrast, we will point out in Section 5 that one can guarantee the Lipschitz continuity of the loss w.r.t input, such as simply by using noise or data augmentation.

4 Sources of instability in GAN

For ease of observing instability sources, consider the vanilla GAN formulation (1) in the case of only 1 real sample x and 1 fake sample $x_z = G(z)$ associated with noise z. Let discriminator $D = D(x; \theta)$ and generator $G = G(z; \phi)$ be parameterized by two neural networks with parameters θ and ϕ, respectively. The loss and its partial derivatives can be written as $V(D, G, x, z, \theta, \phi) = \log D(x) + \log(1 - D(G(z)))$,

$$\frac{\partial V}{\partial x} = \frac{\partial V}{\partial D}\frac{\partial D}{\partial x}, \qquad \frac{\partial V}{\partial z} = \frac{\partial V}{\partial G}\frac{\partial G}{\partial z} = \frac{\partial V}{\partial D}\frac{\partial D}{\partial G}\frac{\partial G}{\partial z} \tag{4}$$

$$\frac{\partial V}{\partial \theta} = \frac{\partial V}{\partial D}\frac{\partial D}{\partial \theta}, \qquad \frac{\partial V}{\partial \phi} = \frac{\partial V}{\partial G}\frac{\partial G}{\partial \phi} = \frac{\partial V}{\partial D}\frac{\partial D}{\partial G}\frac{\partial G}{\partial \phi} \tag{5}$$

4.1 Instability from Lipschitz constraint on one player only

One can observe that the Lipschitz continuity of the loss w.r.t input z depends heavily on the Lipschitz continuity of three functions: the loss V over D, the discriminator D over fake samples $x_z = G(z)$, and the generator G over z. Some existing losses naturally maintain the continuity of V over D, including WGAN, LSGAN, and EBGAN. However, many other losses do not ensure the Lipschitz continuity of V over D, e.g., the vanilla GAN.

Except some recent works (Arjovsky & Bottou, 2017; Guo et al., 2019; Jenni & Favaro, 2019; Qi, 2020), many existing efforts (Arjovsky et al., 2017; Gulrajani et al., 2017; Roth et al., 2017; Fedus et al., 2018; Miyato et al., 2018; Kurach et al., 2019; Zhou et al., 2019; Thanh-Tung et al., 2019; Jiang et al., 2019; Tanielian et al., 2020; Zhao et al., 2020b; Xu et al., 2020; Chu et al., 2020) try to ensure the Lipschitz continuity of the discriminator only. The most popular techniques are gradient

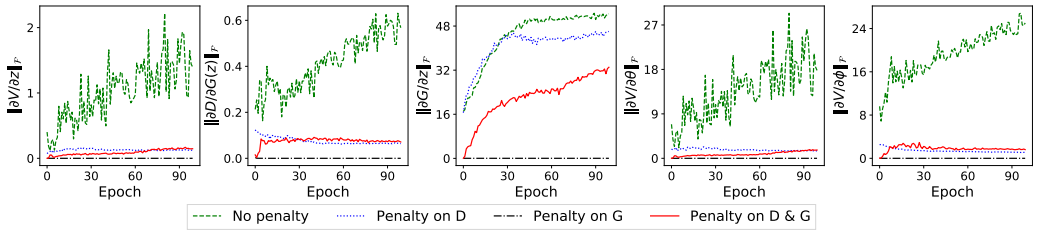

Figure 1: Some statistics when training the vanilla GAN on MNIST. When no penalty was applied, the training algorithm was unstable and unconvergent as shown in the last two subfigures. When spectral normalization was used for G only, the vanishing gradient problem appeared and G did not learn anything from D, and hence the training process stopped early. In contrast, the training behaved very well when spectral normalization was used for D. It can be seen from the middle subfigure that the Lipschitz constant of G can be much smaller if we use spectral normalization for both players. $||\cdot||_{\mathcal{F}}$ denotes the Frobenious norm.

penalty (Gulrajani et al., 2017) and spectral normalization (Miyato et al., 2018; Jiang et al., 2019). Those two techniques are really useful for different losses (Fedus et al., 2018) and high-capacity architectures (Kurach et al., 2019). One reason is that constraining the Lipschitz constant of D indirectly reduces those of the loss w.r.t input z and generator parameter ϕ, as indicated in equations (4) and (5). From a large-scale evaluation, Kurach et al. (2019) found that gradient penalty can help the performance of GANs but does not stabilize the training, whereas using spectral normalization on G only is insufficient to ensure stability (Brock et al., 2019) and can lead to vanishing gradient problem as shown in Figure 1. Such behaviors can be partly explained by using equation (4) and the analysis in the last section. Equation (4) suggests that a Lipschitz constraint on only D (or G) may not ensure the Lipschitz continuity of the loss. Meanwhile, stability basically requires the Lipschitz continuity of the loss w.r.t both the learning algorithm and input (x, z). Existing works mostly focus on input x, but forget input z and the aspect of algorithmic stability (Xu et al., 2010).

Equation (4) suggests that the discriminator should be Lipschitz continuous over both real and fake samples. This supports the use of gradient penalty in (Gulrajani et al., 2017; Thanh-Tung et al., 2019) where D is required to be Lipschitz continuous over the smooth interpolation of real and fake samples. Furthermore, leaving G open may cause inherent difficulties to ensure the Lipschitz continuity of V. As a result, we should simutaneously make sure the Lipschitz continuity of three functions: the loss V over D, the discriminator D over (x, G), and the generator G over z. Figure 1 shows the behaviors of the vanilla GAN in different cases. We observe that without any Lipschitz constraint, both V and D tend to have high Lipschitz constants or be non-smooth, and the training algorithm even did not converge. When spectral normalization is used for both players, those behaviors can be avoided. It is also worth observing that when spectral normalization is applied to G, there is a significant improvement in the Jacobian norm of G.

4.2 Instability from unconvergence

Unconvergent G: It is easy to see that when G is not convergent, the fake samples generated by G are often of low quality, and therefore can be correctly classified by a strong D. This case easily leads to the vanishing gradient problem (Arjovsky & Bottou, 2017) and the generator can hardly learn anything then. The unconvergence of G means $\|\frac{\partial G}{\partial \phi}\|_{\mathcal{F}}$ is large and so $\|\frac{\partial V}{\partial \phi}\|_{\mathcal{F}}$ may be large. This case can happen at the early stage of the learning process, or when we train D until optimality at each training iteration, or when there is no constraint on G as evidenced in Figure 1.

Unconvergent D: This case can happen at the early stage of the training process or in the case of no penalty on D as suggested in Figure 1. In this case, D may not be strong and may make some inaccurate predictions for both real and fake samples. An inaccurate prediction of a fake sample x_z, i.e. $D(x_z) \approx 1$, means that $\frac{\partial V}{\partial x_z} = \frac{-1}{1-D(x_z)}\frac{\partial D}{\partial x_z}$ and $\frac{\partial V}{\partial z} = \frac{\partial V}{\partial x_z}\frac{\partial x_z}{\partial z}$ may have a large norm. Similarly, an inaccurate prediction of a real sample x, i.e. $D(x) \approx 0$, means that $\frac{\partial V}{\partial x} = \frac{1}{D(x)}\frac{\partial D}{\partial x}$ may have a large norm. Some inaccurate predictions can make a significant changes for the updates of both D and G, and may cause mode collapse or gradient explosion (Thanh-Tung et al., 2019). As

Table 1: Some behaviors of GAN when changing the size (B_z) of the noise domain. Spectral normalization was used for D, L_{Vg} was estimated by $\max \|\frac{\partial V_g}{\partial z}\|_{\mathcal{F}}$ from 5000 random noises. The lower the better.

B_z	$L_{Vg}B_z$	loss V_g	$-D(G(z))$
10^{-2}	5.020	-0.657 ± 0.065	-0.480 ± 0.033
10^{0}	2.738	-0.667 ± 0.054	-0.486 ± 0.027
10^{2}	2.370	-0.696 ± 0.045	-0.500 ± 0.022

Table 2: Some behaviors of GAN when changing σ for data augmentation, by random translation. Spectral normalization was used for D, the statistics were estimated from 5000 random noises. The lower the better.

σ	$\sqrt{2}$	$\sqrt{24}$
$\|\frac{\partial D}{\partial G(z)}\|_{\mathcal{F}}$	0.087 ± 0.005	0.068 ± 0.003
$\|\frac{\partial G}{\partial z}\|_{\mathcal{F}}$	78.749 ± 36.405	76.422 ± 35.867
loss V_g	-0.670 ± 0.050	-0.701 ± 0.063
$-D(G(z))$	-0.487 ± 0.026	-0.503 ± 0.031

a result, when stopping the learning process with a unconvergent discriminator, both V and D may have a large Lipschitz constant. Corollary 1 suggests that D and G may not generalize well.

4.3 Instability from the generator input

One issue from noise can be observed from the generalization bound in Corollary 1. The bound will be useful if the domain of z is bounded. This suggests that we should restrict the domain of z when sampling z. However, there is an inherent tradeoff between the size B_z and the Lipschitz constant L_{Vg} of V_g. The first two columns of Table 1 provide an evidence about that tradeoff. We observe further from Table 1 that the generalization of G, indicated by the loss V_g and the score $D(G(z))$, increases as $L_{Vg}B_z$ decreases. Those results consistently support the bound in Corollary 1.

Another possible issue is that a training algorithm for GANs often works with a dynamic environment, for which the training sets for both G and D change over epochs. For example, in the training algorithm by Goodfellow et al. (2014), each minibatch picks randomly a noise sample to put through the generator to make fake inputs for training the discriminator. It turns out that we use different training sets for G over epochs. In other words, the training sets for both G and D are dynamic. Such a dynamic nature may be a source for instability. This partly explains why Shrivastava et al. (2017) propose to keep a history of refined images to train D and to make the overall training more stable. When training from a dynamic dataset, a naive SGD-based method is not guaranteed to converge. Such an unconvergence has been observed by Mescheder et al. (2018), and also can be seen from Figure 1 in the case of no penalty.

5 Why does data augmentation impose a Lipschitz constraint?

In this section, we study a perturbed version of GANs, which allows us to analyze data augmentation. We point out why data augmentation penalizes the norms of the Jacobians of both players, and hence improves stability and generalization for GANs.

Consider the following formulation:

$$\min_G \max_D \mathbb{E}_\epsilon[\mathbb{E}_{x\sim p_d} \log D(x+\epsilon)] + \mathbb{E}_\epsilon[\mathbb{E}_{z\sim p_z} \log(1 - D(G(z)+\epsilon))], \tag{6}$$

where $\epsilon = \sigma u$ and u follows a distribution with mean 0 and covariance matrix I, σ is a non-negative constant. Note that when u is the Gaussian noise, the formulation (6) turns out to be the noisy version of GAN (Arjovsky & Bottou, 2017).

Connection to data augmentation: One difference between (1) and (6) is the input for the discriminator. Note that each input for D in (6) is perturbed by an ϵ. It is easy to see that when ϵ has small norm, each $x' = x + \epsilon$ is a local neighbor of x and is a perturbed version of x. Noise is a common way to make perturbation and can lead to stability for GANs (Arjovsky & Bottou, 2017). Nonetheless, some works (Arjovsky et al., 2017; Zhang et al., 2020) found that using noise often results in G to produce blurry images. Another way to make perturbation is data augmentation, including translation, cutout, rotation. The main idea is to make another version x' from an original image x such that x' should preserve the semantic of x. By this way, x' belongs to the neighborhood of x in some senses, and therefore, can be represented as $x' = x + \epsilon$ for some ϵ. Those observations suggest that when training D and G from a set of original and augmented images (Zhao et al., 2020c; Zhang et al., 2020; Zhao et al., 2020a), we are working with an empirical version of the loss in (6).

Data augmentation penalizes the Jacobian norms: With some abuse of notation, we denote $p_{d+\epsilon}$ be the distribution that generates sample of the form $x+\epsilon$, $p_{g+\epsilon}$ be the distribution that generates sample of the form $G(z)+\epsilon$, and $V(D,G) = \mathbb{E}_\epsilon[\mathbb{E}_{x\sim p_d} \log D(x+\epsilon)] + \mathbb{E}_\epsilon[\mathbb{E}_{z\sim p_z} \log(1-D(G(z)+\epsilon))] = \mathbb{E}_{x\sim p_d}[\mathbb{E}_\epsilon \log D(x+\epsilon)] + \mathbb{E}_{z\sim p_z}[\mathbb{E}_\epsilon \log(1-D(G(z)+\epsilon))]$. Given a fixed G, the optimal discriminator is $D^*(x) = \frac{p_{d+\epsilon}(x)}{p_{d+\epsilon}(x)+p_{g+\epsilon}(x)}$ according to Arjovsky & Bottou (2017). The learning G is to minimize $V(D^*, G)$ given fixed D^*. By using the same argument as Goodfellow et al. (2014), one can see that training G is equivalent to minimizing $\mathbb{E}_\epsilon[JS(p_{d+\epsilon}, p_{g+\epsilon})]$, where JS is the Jensen-Shannon divergence. It is worth observing that

$$\min_G \mathbb{E}_\epsilon[JS(p_{d+\epsilon}, p_{g+\epsilon})] \geq \min_G \mathbb{E}_\epsilon[JS(p_d, p_{g+\epsilon})], \tag{7}$$

$$\min_G \mathbb{E}_\epsilon[JS(p_{d+\epsilon}, p_{g+\epsilon})] \geq \min_G \mathbb{E}_\epsilon[JS(p_{d+\epsilon}, p_g)]. \tag{8}$$

They suggest that training G tries to push $p_{g+\epsilon}$ toward p_d and push p_g toward $p_{d+\epsilon}$. The following results provide some important implications (the proof appears in Appendix B).

Lemma 4 *Let $d(p,q) = \mathbb{E}_{x\sim p_d}[(p(x)-q(x))^2]$ be the squared distance between two functions $p(x)$ and $q(x)$, $J_x(q)$ be the Jacobian of q w.r.t x. Assuming p_d and p_g are differentiable everywhere, then:*
+ $\mathbb{E}_\epsilon[d(p_d, p_{g+\epsilon})] = d(p_d, p_g + o(\sigma)) + \sigma^2 \mathbb{E}_{x\sim p_d}\mathbb{E}_u\left[u^T J_x(p_g) J_x^T(p_g) u\right]$,
+ $\mathbb{E}_\epsilon[d(p_{d+\epsilon}, p_g)] = d(p_d + o(\sigma), p_g) + \sigma^2 \mathbb{E}_{x\sim p_d}\mathbb{E}_u\left[u^T J_x(p_d) J_x^T(p_d) u\right]$.

Lemma 5 *If $u \sim \mathcal{N}(0, I)$ then $\mathbb{E}_u\left[u^T J_x(p_g) J_x^T(p_g) u\right] = trace(J_x(p_g) J_x^T(p_g)) = ||J_x(p_g)||_{\mathcal{F}}^2$ and $\mathbb{E}_u\left[u^T J_x(p_d) J_x^T(p_d) u\right] = trace(J_x(p_d) J_x^T(p_d)) = ||J_x(p_d)||_{\mathcal{F}}^2$.*

Lemma 5 comes from a well-known result (Avron & Toledo, 2011; Hutchinson, 1989) which shows $\mathbb{E}_u\left[u^T A u\right] = trace(A)$ given a fixed matrix A. Similar results hold true for some other types of u, such as Rademacher random variables. Although Lemmas 4 and 5 work with squared distance, they are really helpful to interpret the nontrivial implications when training G and D by (6).

Firstly, when training G, we are trying to minimize the expected norms of the Jacobians of the densities induced by D and G. Indeed, training G will minimize $\mathbb{E}_\epsilon[JS(p_{d+\epsilon}, p_{g+\epsilon})]$, and thus also minimize $\mathbb{E}_\epsilon[JS(p_d, p_{g+\epsilon})]$ according to (7). Because JS is a proper distance, minimizing $\mathbb{E}_\epsilon[JS(p_d, p_{g+\epsilon})]$ leads to minimizing $\mathbb{E}_\epsilon[d(p_d, p_{g+\epsilon})]$. Combining this observation with Lemma 4, we find that training G requires both $d(p_d, p_g + o(\sigma))$ and $\mathbb{E}_{x\sim p_d}\mathbb{E}_u\left[u^T J_x(p_g) J_x^T(p_g) u\right]$ to be small. As a result, $\mathbb{E}_{x\sim p_d}[||J_x(p_g)||_{\mathcal{F}}^2]$ should be small due to Lemma 5. The same argument applies to $\mathbb{E}_{x\sim p_d}[||J_x(p_d)||_{\mathcal{F}}^2]$. Since $D^*(x) = \frac{p_{d+\epsilon}(x)}{p_{d+\epsilon}(x)+p_{g+\epsilon}(x)}$, we can conclude that data augmentation for both real and fake images will implicitly pose Lipschitz constraints on D and G. Furthermore, augmentation for only fake images imposes a Lipschitz constraint on G, while augmentation for only real images imposes a Lipschitz constraint on D.

Secondly, there is a tradeoff in data augmentation. Making augmentation from a larger region around a given image implies a larger σ. Lemmas 4 and 5 suggest that the Jacobian norms should be smaller, meaning the flatter learnt distributions. Hence, too large region for augmentation may result in underfitting. On the other hand, augmentation in a too small region (a small σ) allows the Jacobian norms to be large, meaning the learnt distributions can be more complex. As $\sigma \to 0$, no regularization is used at all. In those cases the generalization of the players may not be guaranteed.

Table 2 shows some statistics from our simulation on MNIST data, where 15 random translations were applied to each image. We observed that increasing σ often results in smaller norms of the Jacobian of both G and D. It seems that a larger σ leads to better fake images as indicated by $D(G(z))$ and V_g on 5000 testing noises. Those results are consistent with our analysis before and the observations by Zhao et al. (2020c).

6 Conclusion

We have discussed the generalization of GANs under more general settings than existing studies, and pointed out a simple way to improve generalization for the players in GANs. Our work provides a theoretically grounded explanation for the highly successful applications of Lipschitz constraints e.g., spectral normalization or gradient penalty. We further suggest data augmentation to be an effective alternative which is extremely cheap in practice.

REFERENCES

Martin Arjovsky and Leon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.

Martin Arjovsky, Soumith Chintala, and Leon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *International Conference on Machine Learning*, pp. 224–232, 2017.

Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2):1–34, 2011.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Tatjana Chavdarova, Matteo Pagliardini, Martin Jaggi, and Francois Fleuret. Taming gans with lookahead. *arXiv preprint arXiv:2006.14567*, 2020.

Casey Chu, Kentaro Minami, and Kenji Fukumizu. Smoothness and stability in gans. In *International Conference on Learning Representations*, 2020.

William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. In *International Conference on Learning Representations*, 2018.

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 C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

Tianyu Guo, Chang Xu, Boxin Shi, Chao Xu, and Dacheng Tao. Smooth deep image generator from noises. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3731–3738, 2019.

Tianyu Guo, Chang Xu, Jiajun Huang, Yunhe Wang, Boxin Shi, Chao Xu, and Dacheng Tao. On positive-unlabeled classification in gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8385–8393, 2020.

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, 2017.

Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo, and Sungroh Yoon. How generative adversarial networks and their variants work: An overview. *ACM Computing Surveys (CSUR)*, 52(1):1–43, 2019.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18(3):1059–1076, 1989.

Simon Jenni and Paolo Favaro. On stabilizing generative adversarial training with noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12145–12153, 2019.

Haoming Jiang, Zhehui Chen, Minshuo Chen, Feng Liu, Dingding Wang, and Tuo Zhao. On computation and generalization of generative adversarial networks under spectrum control. In *International Conference on Learning Representations*, 2019.

Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. In *International Conference on Learning Representations*, 2019.

Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7799–7808, 2020.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. 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.

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

Karol Kurach, Mario Lučić, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in gans. In *International Conference on Machine Learning*, pp. 3581–3590, 2019.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pp. 2203–2213, 2017.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in Neural Information Processing Systems*, pp. 700–709, 2018.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, 2017.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. On the effectiveness of least squares generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):2947–2960, 2019.

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. 3481–3490, 2018.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Vaishnavh Nagarajan and J Zico Kolter. Gradient descent gan optimization is locally stable. In *Advances in Neural Information Processing Systems*, pp. 5585–5595, 2017.

Weili Nie and Ankit Patel. Towards a better understanding and regularization of gan training dynamics. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.

Guo-Jun Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *International Journal of Computer Vision*, 128(5):1118–1140, 2020.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016.

Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems*, pp. 2018–2028, 2017.

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. 2234–2242, 2016.

Maziar Sanjabi, Jimmy Ba, Meisam Razaviyayn, and Jason D Lee. On the convergence and robustness of training gans with regularized optimal transport. In *Advances in Neural Information Processing Systems*, pp. 7091–7101, 2018.

Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8207–8216, 2020.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2107–2116, 2017.

Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jeremie Mary. Learning disconnected manifolds: a no gan's land. In *International Conference on Machine Learning*, 2020.

Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. Improving generalization and stability of generative adversarial networks. In *International Conference on Learning Representations*, 2019.

Dong Wang, Xiaoqian Qin, Fengyi Song, and Li Cheng. Stabilizing training of generative adversarial nets via langevin stein variational gradient descent. *arXiv preprint arXiv:2004.10495*, 2020.

Bingzhe Wu, Shiwan Zhao, Chaochao Chen, Haoyang Xu, Li Wang, Xiaolu Zhang, Guangyu Sun, and Jun Zhou. Generalization in generative adversarial networks: A novel perspective from privacy protection. In *Advances in Neural Information Processing Systems*, pp. 307–317, 2019.

Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robust regression and lasso. *IEEE Transactions on Information Theory*, 56(7):3561–3574, 2010.

Kun Xu, Chongxuan Li, Huanshu Wei, Jun Zhu, and Bo Zhang. Understanding and stabilizing gans' training dynamics with control theory. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Yasin Yazıcı, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, and Vijay Chandrasekhar. The unusual effectiveness of averaging in gan training. In *International Conference on Learning Representations*, 2019.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pp. 7354–7363, 2019.

Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations*, 2020.

Pengchuan Zhang, Qiang Liu, Dengyong Zhou, Tao Xu, and Xiaodong He. On the discrimination-generalization tradeoff in gans. In *International Conference on Learning Representations*, 2018.

Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial networks. In *International Conference on Learning Representations*, 2017.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *arXiv preprint arXiv:2006.10738*, 2020a.

Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans. *arXiv preprint arXiv:2002.04724*, 2020b.

Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020c.

Zhiming Zhou, Jiadong Liang, Yuxuan Song, Lantao Yu, Hongwei Wang, Weinan Zhang, Yong Yu, and Zhihua Zhang. Lipschitz generative adversarial nets. In *International Conference on Machine Learning*, pp. 7584–7593, 2019.

A LOCAL LINEARITY

Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ which is differentiable everywhere in its domain. f is also called locally linear everywhere. Let $\epsilon = \sigma u$, where u follows a distribution with mean 0 and covariance matrix I, $\sigma \geq 0$, $J_x(f)$ be the Jacobian of f w.r.t the input of f. Considering $f(x+\epsilon) = f(x+\sigma u)$ as a function of σ, Taylor's theorem allows us to write $f(x + \sigma u) = f(x) + \sigma J_x^T(f)u + o(\sigma)$. Therefore,

$$\begin{aligned}
\mathbb{E}_\epsilon[f(x+\epsilon)] &= \mathbb{E}_\epsilon[f(x) + \sigma J_x^T(f)u + o(\sigma)] && (9)\\
&= f(x) + o(\sigma) + \sigma \mathbb{E}_u[J_x^T(f)u] && (10)\\
&= f(x) + o(\sigma), && (11)
\end{aligned}$$

where we have used $\mathbb{E}_u[J_x^T(f)u] = 0$ due to $\mathbb{E}_u[u] = 0$ and the independence of the elements of u. As $\sigma \to 0$, we have $\mathbb{E}_\epsilon[f(x+\epsilon)] \to f(x)$. In other words, when σ is sufficiently small, one can well approximate $f(x) = \mathbb{E}_\epsilon[f(x+\epsilon)]$ at any x.

B JACOBIAN CONSTRAINT ON G AND D

Lemma 6 *Let $d(p,q) = \mathbb{E}_{x\sim p_d}[(p(x)-q(x))^2]$ be the squared distance between two functions $p(x)$ and $q(x)$, $J_x(q)$ be the Jacobian of q w.r.t x. Assuming p_d and p_g are differentiable everywhere, then:*
+ $\mathbb{E}_\epsilon[d(p_d, p_{g+\epsilon})] = d(p_d, p_g + o(\sigma)) + \sigma^2 \mathbb{E}_{x\sim p_d}\mathbb{E}_u\left[u^T J_x(p_g) J_x^T(p_g) u\right]$,
+ $\mathbb{E}_\epsilon[d(p_{d+\epsilon}, p_g)] = d(p_d + o(\sigma), p_g) + \sigma^2 \mathbb{E}_{x\sim p_d}\mathbb{E}_u\left[u^T J_x(p_d) J_x^T(p_d) u\right]$,
where $o(\sigma)$ denotes a function satisfying $\lim_{\sigma\to 0} o(\sigma)/\sigma = 0$.

Proof: Due to the everywhere differentiability of p_g, we have $p_{g+\epsilon} - p_g = \sigma J_x^T(p_g)u + o(\sigma)$ and $\mathbb{E}_\epsilon[p_{g+\epsilon} - p_g] = o(\sigma)$ as analyzed in Appendix A. Next, we consider

$$\begin{aligned}
\mathbb{E}_\epsilon[d(p_d, p_{g+\epsilon})] &= \mathbb{E}_\epsilon\mathbb{E}_{x\sim p_d}[(p_d - p_{g+\epsilon})^2] = \mathbb{E}_\epsilon\mathbb{E}_{x\sim p_d}[(p_d - p_g + p_g - p_{g+\epsilon})^2]\\
&= \mathbb{E}_\epsilon\mathbb{E}_{x\sim p_d}[(p_d - p_g)^2] + 2\mathbb{E}_\epsilon\mathbb{E}_{x\sim p_d}[(p_d - p_g)(p_g - p_{g+\epsilon})] + \mathbb{E}_\epsilon\mathbb{E}_{x\sim p_d}[(p_g - p_{g+\epsilon})^2]\\
&= \mathbb{E}_{x\sim p_d}[(p_d - p_g)^2] + 2\mathbb{E}_{x\sim p_d}\mathbb{E}_\epsilon[(p_d - p_g)(p_g - p_{g+\epsilon})] + \mathbb{E}_\epsilon\mathbb{E}_{x\sim p_d}[(p_g - p_{g+\epsilon})^2]\\
&= \mathbb{E}_{x\sim p_d}[(p_d - p_g)^2] - 2o(\sigma)\mathbb{E}_{x\sim p_d}[(p_d - p_g)] + \mathbb{E}_{x\sim p_d}\mathbb{E}_\epsilon[(p_g - p_{g+\epsilon})^2]\\
&= \mathbb{E}_{x\sim p_d}[(p_d - p_g - o(\sigma))^2] - o(\sigma^2) + \mathbb{E}_{x\sim p_d}\mathbb{E}_\epsilon[(p_g - p_{g+\epsilon})^2] && (12)\\
&= d(p_d, p_g + o(\sigma)) - o(\sigma^2) + \mathbb{E}_{x\sim p_d}\mathbb{E}_\epsilon[(p_g - p_{g+\epsilon})^2]. && (13)
\end{aligned}$$

Note that

$$\begin{aligned}
\mathbb{E}_\epsilon[(p_g - p_{g+\epsilon})^2] &= \mathbb{E}_\epsilon\left[\sigma J_x^T(p_g)u + o(\sigma)\right]^2 && (14)\\
&= \mathbb{E}_\epsilon\left[\sigma^2 u^T J_x(p_g) J_x^T(p_g) u + 2\sigma o(\sigma) J_x^T(p_g)u + o(\sigma^2)\right] && (15)\\
&= \sigma^2 \mathbb{E}_u\left[u^T J_x(p_g) J_x^T(p_g) u\right] + o(\sigma^2). && (16)
\end{aligned}$$

Replacing (16) to equation (13) will lead to the first statement. The second statement can be shown by using the same argument, completing the proof.

C EXPERIMENTAL SETUPS

The architectures of G and D are specified in Figure 2, which follow `http://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/gan/gan.py`.

We use MNIST dataset which has 60000 images for training and 5000 images for testing. During the testing phase, 5000 new noises are sampled randomly at every epoch or for computing some metrics. Before fetching into D, both real and fake images are converted to tensor size $(1, 28, 28)$, rescaled to $(0, 1)$ and normalized with $mean = 0.5$ and $std = 0.5$. The noise input of G has 100 dimensions and is sampled from either uniform or normal distribution. We use Adam optimizer with $\beta_1 = 0.5, \beta_2 = 0.999, lr = 0.0002$, $batchsize = 64$ and $epochs = 200$.

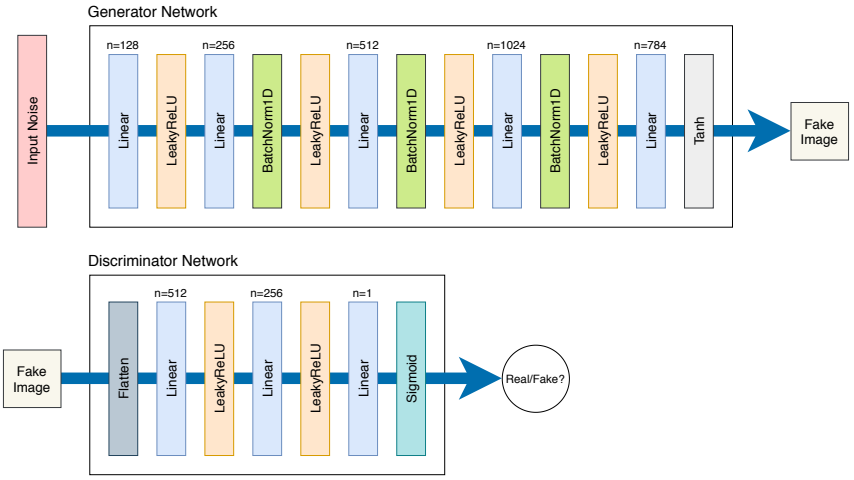

Figure 2: The architectures of G and D with the negative slope of `LeakyRuLU` is 0.2

D CHANGING B_z

In this experiment, the input noise of G is sampled from uniform distribution $\mathcal{U}[-s, s]$ for $s \in \{0.01, 1, 100\}$. Spectral normalization (Miyato et al., 2018) is used for D. Other setups are the same as Appendix C. Figure 3 shows the results along the training progress.

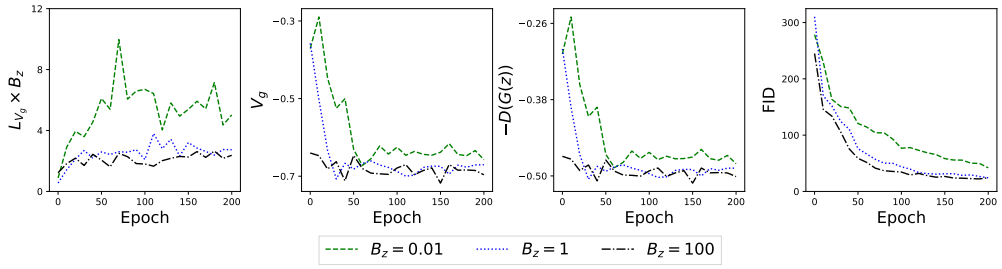

Figure 3: Some behaviors of GAN when changing B_z, the lower the better. L_{V_g} was estimated by $\max \|\frac{\partial V_g}{\partial z}\|_{\mathcal{F}}$ from 5000 testing noises, while the other quantities were averaged from from 5000 testing noises. We can see that $B_z = 0.01$ provides the worst results, while the results for $B_z = 1$ are slightly worse than those for $B_z = 100$. Those results are consistent with our theoretical analysis.

E CHANGING σ

In this experiment, the input noise of G is sampled from $\mathcal{N}(0, I)$. For the input of D, both real and fake images are augmented randomly 15 times, then the augmented images and the original images are fetched into D. We only use image translation for augmentation. The shifts in horizontal and vertical axis are sampled from discrete uniform distribution within interval $[-s, s]$, where $s = 2$ corresponds to $\sigma = \sqrt{2}$ and $s = 8$ corresponds to $\sigma = \sqrt{24}$. Spectral normalization (Miyato et al., 2018) is used for D. Other setups are the same as Appendix C. Figure 4 shows the results along the training progress.

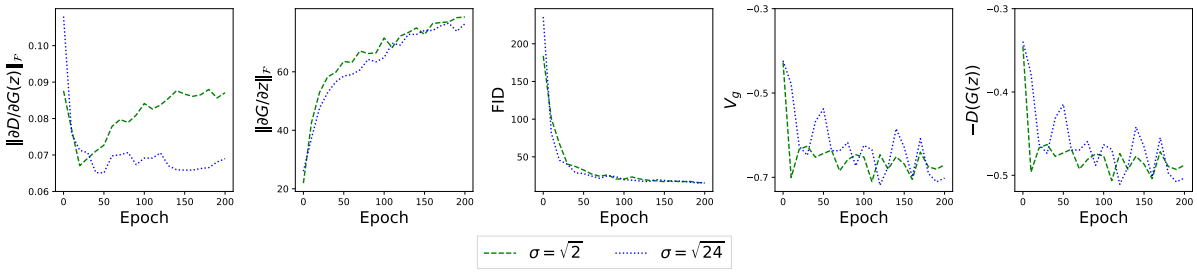

Figure 4: Some behaviors of GAN with different σ for augmentation. It can be seen from the first two subfigures that the higher σ provides smaller Jacobian norms. This is consistent with our theoretical analysis. However, the last three subfigures suggest that there is no clear difference between the two settings of σ.