# Towards a Better Understanding and Regularization of GAN Training Dynamics

**Weili Nie**
Rice University
wn8@rice.edu

**Ankit B. Patel**
Rice University & Baylor College of Medicine
abp4@rice.edu

## Abstract

Generative adversarial networks (GANs) are notoriously difficult to train and the reasons underlying their (non-)convergence behaviors are still not completely understood. By first considering a simple yet representative GAN example, we mathematically analyze its local convergence behavior in a non-asymptotic way. Furthermore, the analysis is extended to general GANs under certain assumptions. We find that in order to ensure a good convergence rate, two factors of the Jacobian in the GAN training dynamics should be *simultaneously* avoided, which are (i) the *Phase Factor*, i.e., the Jacobian has complex eigenvalues with a large imaginary-to-real ratio, and (ii) the *Conditioning Factor*, i.e., the Jacobian is ill-conditioned. Previous methods of regularizing the Jacobian can only alleviate one of these two factors, while making the other more severe. Thus we propose a new *JA*cobian *RE*gularization (JARE) for GANs, which simultaneously addresses both factors by construction. Finally, we conduct experiments that confirm our theoretical analysis and demonstrate the advantages of JARE over previous methods in stabilizing GANs.

## 1 INTRODUCTION

Generative adversarial networks (GANs) (Goodfellow et al., 2014) have achieved great success at generating realistic samples, with extensive applications (Ho and Ermon, 2016; Zhu et al., 2017; Karras et al., 2019). The goal of GANs is to generate samples that are indistinguishable from real data and hence have essentially learned the underlying data distribution. However, they are notoriously difficult to train and as such many heuristics have been developed (Radford et al., 2015; Salimans et al., 2016; Brock et al., 2019). Meanwhile, a lot of theoretical work has focused on stabilizing the GAN training by replacing the Jensen-Shannon (JS) divergence implicit in the vanilla GAN (Goodfellow et al., 2014) with alternative divergences, such as $f$-divergence (i.e. $f$-GAN) (Nowozin et al., 2016) and Wasserstein distance (i.e. WGAN) (Arjovsky et al., 2017). Much of the related work has introduced various regularizers for better approximating these divergences (Gulrajani et al., 2017; Roth et al., 2017; Miyato et al., 2018).

But the training dynamics of GANs are still not completely understood. Typically, the training of GANs is achieved by solving a zero-sum game via simultaneous gradient descent (SimGD) (Goodfellow et al., 2014; Nowozin et al., 2016; Arjovsky et al., 2017). The original work (Goodfellow et al., 2014) showed that SimGD converges to an equilibrium if the updates are made in the function space. In practice, with the generator and discriminator being parametrized by two distinct neural networks, the updates in the parameter space are no longer guaranteed to converge due to the highly non-convex properties of the loss surface (Goodfellow, 2016).

In this work, we conduct a non-asymptotic analysis of local convergence in GAN training dynamics by evaluating the eigenvalues of its Jacobian near the equilibrium and analyzing the convergence rate. We first consider a simple yet representative GAN example, and then extend the analysis to the general GANs, where we find that the number of iterations needed to achieve $\epsilon$-error may be unexpectedly large due to the *Phase Factor* (i.e., the Jacobian has complex eigenvalues with a large imaginary-to-real ratio) and *Conditioning Factor* (i.e., the Jacobian is ill-conditioned) of the Jacobian. We later show that previous methods of regularizing the Jacobian can only alleviate one of these two factors, while making the impact of the other factor more severe. Based on our analysis, we propose a new *JA*cobian *RE*gularization (JARE)

for GANs and show theoretically that it can alleviate these two factors simultaneously. Finally, experiments confirm our theoretical analysis and demonstrate the advantages of JARE over recently proposed methods.

## 2  RELATED WORK

**Global convergence of GANs.**  By assuming the GAN objectives to be convex-concave, many works have provided the global convergence behaviors of GANs (Nowozin et al., 2016; Yadav et al., 2018; Gidel et al., 2019). However, as shown in Section 4, the convex-concave assumption is too unrealistic to hold true even in a simple GAN example. Also, Li et al. (2018) showed the global convergence of GANs by assuming a parametrized mixture of two Gaussians as the generator. Nevertheless, their theoretical results only work for GANs provided an optimal discriminator. These unrealistic assumptions together make an inevitably large gap between their theory and the actual training dynamics of GANs. Instead, we focus on the local convergence of GANs, which is a necessary condition of global convergence but more analytically tractable as it eschews such strong assumptions.

**Local convergence of GANs.**  Recently, Nagarajan and Kolter (2017) showed that under some mild assumptions, the GAN dynamics are locally convergent. Furthermore, Mescheder et al. (2018) pointed out that if the assumptions in Nagarajan and Kolter (2017) are not satisfied, in particular when data distributions are not absolutely continuous, the GAN dynamics are not always convergent, unless some regularization techniques are applied, such as zero-centered gradient penalties (Roth et al., 2017) and consensus optimization (ConOpt) (Mescheder et al., 2017). However, these theoretical results are established in an asymptotic limit of vanishing step size where SimGD approximates a continuous-time dynamic system. In practice, we are more concerned about the characterization of the non-asymptotic convergence rate and the choice of the finite step size. This is because even though the continuous-time dynamic system is convergent, its discrete-time counterpart might still suffer from a poor convergence behavior. To this end, Liang and Stokes (2019) analyzed the non-asymptotic local convergence of GANs and revealed that the off-diagonal interaction term in the Jacobian can serve as both a blessing and a curse. Our theoretical results can serve a complementary to Liang and Stokes (2019) in terms of better understanding the local convergence of GANs.

**General differentiable games.**  Another line of related work has focused on analyzing the general differentiable games with GANs being a specific use case (Balduzzi et al., 2018; Daskalakis and Panageas, 2018; Letcher et al., 2019). In particular, Balduzzi et al. (2018) decomposed the game dynamics into two components and proposed the Symplectic Gradient Adjustment (SGA) to find stable fixed points in general games. Interestingly, SGA shares some similarities with JARE in form although we are motivated from completely different perspectives. The major difference between SGA and JARE is that SGA needs an exclusive sign alignment during training which JARE does not require, and we argue that a better understanding and improvement of GAN dynamics should take the GAN properties into account, which is missing in this line of related work.

## 3  BACKGROUND

### 3.1  GAN AS A MINIMAX GAME

Despite many variants, the GAN is best described as a minimax game in which the two players, usually named the generator and discriminator, are maximizing and minimizing the same objective function, respectively. The GAN game can be formulated as follows:

$$\min_{\phi} \max_{\theta} \ f(\phi, \theta)$$

$$f(\phi, \theta) \triangleq \mathbb{E}_{x \sim P_r}[g_1(D_\theta(x))] + \mathbb{E}_{z \sim P_0}[g_2(D_\theta(G_\phi(z)))] \tag{1}$$

where $\phi \in \Phi \subseteq \mathbb{R}^m$ and $\theta \in \Theta \subseteq \mathbb{R}^n$ denote the parameters of the generator $G_\phi : \mathcal{Z} \to \mathcal{X}$ and discriminator $D_\theta : \mathcal{X} \to \mathbb{R}$, respectively, $P_r$ and $P_0$ represent the true data distribution with support $\mathcal{X} \subseteq \mathbb{R}^d$ and latent distribution with support $\mathcal{Z}$. We also denote by $P_\phi$ the generated data distribution. Note that in our definition, the output of the discriminator $D_\theta$ is a real-valued logit rather than a probability. Therefore, by relating the objective in (1) to different $f$-divergences and Wasserstein distance between $P_r$ and $P_\phi$, $g_1, g_2 : \mathbb{R} \to \mathbb{R}$ are both *concave* functions, which is similar to Nagarajan and Kolter (2017). For example, we can recover vanilla GAN with $g_1(t) = g_2(-t) = -\log(1 + e^{-t})$, WGAN with $g_1(t) = g_2(-t) = t$ and reverse Kullback-Leibler (KL) divergence in $f$-GAN with $g_1(t) = -e^{-t}, g_2(t) = 1 - t$[1].

For training the minimax GAN game (1), SimGD is the most commonly used algorithm, in which the parameter updates are alternatively given as

$$\phi^{(k+1)} = \phi^{(k)} - \eta \nabla_\phi f(\phi^{(k)}, \theta^{(k)})$$
$$\theta^{(k+1)} = \theta^{(k)} + \eta \nabla_\theta f(\phi^{(k)}, \theta^{(k)}) \tag{2}$$

---

[1]Normally, WGAN requires the discriminator parameter space $\Theta$ to be an $K$-Lipschitz functional space while for $f$-divergences, we can simply set $\Theta = \mathbb{R}^n$.

where $\eta > 0$ is the step size, $\phi^{(k)}$ and $\theta^{(k)}$ denote the corresponding parameters in the $k$-th iteration. Due to the non-convex properties of the GAN objective (Goodfellow, 2016), it is difficult to analyze its global convergence in general. To gain key insights into the training instabilities in GANs, we focus on the local convergence of points near the equilibrium (Nagarajan and Kolter, 2017; Mescheder et al., 2018, 2017; Heusel et al., 2017).

## 3.2 ASYMPTOTIC VS. NON-ASYMPTOTIC CONVERGENCE ANALYSIS

The asymptotic convergence analysis is defined as applying the "ordinary differential equation (ODE) method" to analyze the convergence properties of dynamic systems. For example, consider a discrete-time system characterized by the gradient descent $v^{(t+1)} = v^{(t)} + \eta h(v^{(t)})$ for the gradient $h(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ and step size $\eta > 0$, the asymptotic convergence analysis assumes the step size $\eta$ is infinitely small such that the discrete-time system can be approximated by an ODE $\dot{v}(t) = h(v(t))$. According to the Linearization Theorem (Arrowsmith and Place, 1992), if the Jacobian of the dynamic system $A \triangleq \frac{\partial h(v)}{\partial v}$ evaluated at a stationary point $v^*$ is Hurwitz, namely, $\mathrm{Re}\{\lambda_i(A)\} < 0, \forall i = 1, \cdots, n$, the equivalent ODE will converge to $v^*$ for all points in its neighborhood.

In the non-asymptotic convergence analysis, however, we consider the discrete system directly to obtain the number of iterations needed to achieve an $\epsilon$-error solution with a finite step size. Particularly, given the Jacobian $A$, to ensure the non-asymptotic convergence, we first provide an appropriate range of step size $\eta$ by solving the inequalities $|1 + \lambda_i(A)| < 1, \forall i = 1, \cdots, n$. Based on the constraint of the step size, we get the minimum value of $|1 + \lambda_i(A)|$, and thus are able to evaluate the minimum number of iterations for an $\epsilon$-error solution, which characterizes the convergence rate. Therefore, the non-asymptotic analysis could more precisely reveal the convergence performance of the dynamic system than the asymptotic analysis.

## 4  A SIMPLE GAN EXAMPLE

For illustration, we first consider a simple GAN example, in which the true data distribution is an isotropic Gaussian with a nonzero mean, i.e. $x \sim \mathcal{N}(v, \sigma^2 I)$ where $x \in \mathbb{R}^n$ (assuming $d = n$) and latent distribution is also a Gaussian with the same shape but a zero mean, i.e. $z \sim \mathcal{N}(0, \sigma^2 I)$ where $z \in \mathbb{R}^n$. Basically, the problem becomes whether the generator could translate the latent Guassian to match the real Gaussian. To this end, we can assume the generator and discriminator are both *linear*, i.e. $D_\theta(x) = \theta^T x$ (assuming $m = n$) and $G_\phi(z) = \phi + z$, which are both provably expressive

enough to learn the true data distribution. Thus, the GAN game objective in (1) can be rewritten as

$$
\begin{aligned}
f(\phi, \theta) =& \mathbb{E}_{x \sim \mathcal{N}(v, \sigma^2 I)}[g_1(\theta^T x)] \\
& + \mathbb{E}_{z \sim \mathcal{N}(0, \sigma^2 I)}[g_2(\theta^T (\phi + z))]
\end{aligned}
\tag{3}
$$

It is easy to verify that the equilibrium exists, which is $(\phi^*, \theta^*) = (v, 0)$. Before proceeding to the analysis, we show that this simple GAN example is in fact a *concave-concave* game, essentially different from the previous convex-concave assumption in GANs (Nowozin et al., 2016; Yadav et al., 2018; Gidel et al., 2019).

**Lemma 1.** *The objective $f(\phi, \theta)$ in (3) is concave-concave w.r.t. $(\phi, \theta)$.*

*Proof:* See Appendix A.1. $\qquad\square$

By considering a small open neighborhood of $(\phi^*, \theta^*)$ of radius $\delta$, denoted by $B_\delta(\phi^*, \theta^*)$, we introduce the local properties in this simple GAN example as follows.

**Lemma 2.** *The second-order derivative of $f(\phi, \theta)$ in (3) w.r.t. $(\phi, \theta) \in B_\delta(\phi^*, \theta^*)$ is given by*

$$
\begin{aligned}
\nabla^2 f(\phi, \theta) &\triangleq \begin{bmatrix} \nabla^2_{\phi\phi} f(\phi, \theta) & \nabla^2_{\phi\theta} f(\phi, \theta) \\ \nabla^2_{\theta\phi} f(\phi, \theta) & \nabla^2_{\theta\theta} f(\phi, \theta) \end{bmatrix} \\
&\approx \begin{bmatrix} 0 & g_2'(0)I \\ g_2'(0)I & (g_1''(0) + g_2''(0))(\sigma^2 I + vv^T) \end{bmatrix}
\end{aligned}
\tag{4}
$$

*Proof:* See Appendix A.2. $\qquad\square$

Without loss of generality, we focus on the vanilla GAN objective, i.e. $g_1(t) = g_2(-t) = -\log(1 + e^{-t})$, in the rest of the paper, since the analysis in general applies to different GAN objectives. To simplify notations, we let $w \triangleq (\phi - v, \theta)$ so the equilibrium becomes $w^* = 0$ and the SimGD updates in (2) can be rewritten as

$$
w^{(k+1)} = w^{(k)} + \eta \tilde{\nabla} f(w^{(k)})
\tag{5}
$$

where $\tilde{\nabla} f(w^{(k)}) \triangleq \begin{bmatrix} -\nabla_\phi f(w^{(k)}) \\ \nabla_\theta f(w^{(k)}) \end{bmatrix}$, and thus the Jacobian at $w^{(k)}$ is given by

$$
A(w^{(k)}) \triangleq \frac{\partial \tilde{\nabla} f(w^{(k)})}{\partial w^{(k)}}^T = \begin{bmatrix} -\nabla^2_{\phi\phi} f(w^{(k)}) & -\nabla^2_{\phi\theta} f(w^{(k)}) \\ \nabla^2_{\theta\phi} f(w^{(k)}) & \nabla^2_{\theta\theta} f(w^{(k)}) \end{bmatrix}
$$

In the next, we will replace $A(w^{(k)})$ by $A$ for brevity.

**Theorem 1.** *For any point within $B_\delta(w^*)$, the Jacobian $A$ in the simple vanilla GAN example trained via SimGD has the following eigenvalues: $\lambda_{1,2}(A) = \frac{-\sigma^2 \pm \sqrt{(\sigma^2)^2 - 4}}{4}$ and $\lambda_{3,4}(A) = \frac{-\beta^2 \pm \sqrt{(\beta^2)^2 - 4}}{4}$ where $\beta^2 \triangleq \sigma^2 + \|v\|^2$.*

*Proof:* See Appendix A.3. □

The above theorem shows that $\mathrm{Re}\{\lambda_{1,2}(A)\} < 0$ and $\mathrm{Re}\{\lambda_{3,4}(A)\} < 0$, and thus the SimGD updates in this simple GAN example are asymptotically locally convergent, which is consistent with Nagarajan and Kolter (2017). Next, we discuss lower bounds of the non-asymptotic convergence rate in two cases.

On the one hand, assuming the variance satisfies $0 < \sigma^2 < 2$, $\lambda_{1,2}(A)$ become complex-valued. Denote by $\zeta \triangleq \left| \frac{\mathrm{Im}\{\lambda_{1,2}(A)\}}{\mathrm{Re}\{\lambda_{1,2}(A)\}} \right|$ the absolute value of the imaginary-to-real ratio of $\lambda_{1,2}(A)$. The non-asymptotic convergence property determined by $\lambda_{1,2}(A)$ is given as follows.

**Corollary 1.** *To ensure non-asymptotic local convergence, the step size should satisfy $0 < \eta < \frac{4}{\sqrt{1+\zeta^2}}$. The number of iterations to achieve an $\epsilon$-error solution satisfies $N \geq \frac{2\log\frac{C_0}{\epsilon}}{\log(1+\frac{1}{\zeta^2})}$ where $C_0$ is a constant. Specifically, as $\zeta \to \infty$, $N$ will be at least $O\left(\zeta^2 \log\frac{1}{\epsilon}\right)$.*

*Proof:* See Appendix A.4. □

It means when the absolute value of the imaginary-to-real ratio of $\lambda_{1,2}(A)$ increases, the number of iterations $N$ for a certain convergence performance increases (quadratically in the limit). Since we know $\zeta = \sqrt{(\frac{2}{\sigma^2})^2 - 1}$ in the simple vanilla GAN example, which is a monotonically decreasing function of $\sigma^2$, if we set $\sigma^2 = 0.01$ for instance, then $N \geq O(10^4 \log\frac{1}{\epsilon})$ which shows a quite slow convergence rate.

On the other hand, we assume $\beta^2 > 2$, then $\lambda_{3,4}(A)$ are real-valued. Without loss of generality, we assume $|\lambda_3(A)| \geq |\lambda_4(A)|$ and the absolute value of their ratio is denoted by $\tau \triangleq \left| \frac{\lambda_3(A)}{\lambda_4(A)} \right|$. Thus, $\tau$ is a lower bound of the condition number of the Jacobian, and the non-asymptotic convergence property determined by $\lambda_{3,4}(A)$ is given as follows.

**Corollary 2.** *To ensure non-asymptotic local convergence, the step size should also satisfy $0 < \eta < \frac{4}{\sqrt{\tau}}$. For $\tau > 2$, the number of iterations $N$ to achieve an $\epsilon$-error solution satisfies $N > \frac{\log\frac{\epsilon}{C_1}}{\log(1-\frac{2}{\tau})}$ where $C_1$ is a constant. Specifically, as $\tau \to \infty$, $N$ will be at least $O(\tau \log\frac{1}{\epsilon})$.*

*Proof:* See Appendix A.5. □

It means when the absolute value of $\frac{\lambda_3(A)}{\lambda_4(A)}$ increases, the number of iterations $N$ for a certain convergence performance also increases (linearly in the limit). Since we know $\tau = \frac{1}{4}(\beta^2 + \sqrt{(\beta^2)^2 - 4})^2$ in the simple vanilla GAN example, which is a monotonically increasing function of $\beta^2$, if we set $\|v\| = 10$ for instance, then $N \geq O(10^4 \log\frac{1}{\epsilon})$, which also implies a very poor con-

vergence rate.

In summary, there may exist the following two factors of the Jacobian in the GAN dynamics *simultaneously* (e.g., $0 < \sigma^2 < 2$ and $\beta^2 > 2$ in the simple GAN example) that result in the GAN training issues.

- *Phase Factor*: The Jacobian $A$ has complex eigenvalues with a large imaginary-to-real ratio, which has also been reported in Mescheder et al. (2017).

- *Conditioning Factor*: The Jacobian $A$ is ill-conditioned, i.e., the largest absolute value of its eigenvalues is much larger than the smallest one.

As we can see later in general GANs, it is the special nature of the Jacobian in GANs that makes the GAN training dynamics more unstable than other neural network optimization problems. In particular, Theorem 1 reveals that in the simple GAN example, both $\sigma^2$ and $\beta^2$ should not be too small or too large, which is a relatively strict requirement for local convergence. Furthermore, simply changing the expressive power of the generator or discriminator may not easily alleviate these two factors simultaneously. Please see Appendix B for an example of changing the discriminator representations. Therefore, how to simultaneously alleviate these two factors we have identified above becomes an important question for the GAN training.

## 5 JACOBIAN REGULARIZATION

A straightforward method to alleviate these two factors simultaneously is to introduce a *regularization matrix* $\Gamma$ such that the training updates in (5) become

$$w^{(k+1)} = w^{(k)} + \eta\Gamma\tilde{\nabla}f(w^{(k)}) \qquad (6)$$

and thus the (regularized) Jacobian is given by $A = \Gamma\frac{\partial\tilde{\nabla}f(w^{(k)})}{\partial w^{(k)}}^T$. The goal is to find a regularization matrix $\Gamma$ such that we can appropriately control the eigenvalues of the Jacobian for points near the equilibrium.

### 5.1 REVISITING PREVIOUS METHODS

There are several gradient-based regularization methods that have been proposed to deal with the training instabilities of GANs from the perspective of controlling the Jacobian such as *only regularizing generator* (Nagarajan and Kolter, 2017) and *ConOpt* (Mescheder et al., 2017).

**Only regularizing generator.** To overcome the non-convergence issue of training WGAN via SimGD, Nagarajan and Kolter (2017) has proposed to only regularize the generator by using the gradient of the discriminator in a principled way. The regularized updates for the

generator become

$$\phi^{(k+1)} = \phi^{(k)} - \eta\nabla_\phi f(w^{(k)}) - \frac{1}{2}\eta\gamma\nabla_\phi \left\|\nabla_\theta f(w^{(k)})\right\|^2$$

where the discriminator updates remain the same with SimGD, and thus the corresponding regularization matrix is $\Gamma = \begin{bmatrix} I & -\gamma\nabla^2_{\phi\theta}f(w^{(k)}) \\ 0 & I \end{bmatrix}$ with $\gamma$ being a tunable hyperparameter.

**ConOpt.** By directly alleviating the impact of the *Phase Factor*, Mescheder et al. (2017) has proposed ConOpt and its regularized updates are

$$w^{(k+1)} = w^{(k)} + \eta\tilde{\nabla}f(w^{(k)}) - \frac{1}{2}\eta\gamma\nabla\left\|\nabla f(w^{(k)})\right\|^2$$

where the corresponding regularization matrix is $\Gamma = \begin{bmatrix} I + \gamma\nabla^2_{\phi\phi}f(w^{(k)}) & -\gamma\nabla^2_{\phi\theta}f(w^{(k)}) \\ \gamma\nabla^2_{\theta\phi}f(w^{(k)}) & I - \gamma\nabla^2_{\theta\theta}f(w^{(k)}) \end{bmatrix}$.

**Only regularizing discriminator.** Similar to *only regularizing generator* in (Nagarajan and Kolter, 2017), a straightforward idea is to only regularize the discriminator instead by using the gradient of the generator and its regularized updates for the discriminator become

$$\theta^{(k+1)} = \theta^{(k)} + \eta\nabla_\theta f(w^{(k)}) - \frac{1}{2}\eta\gamma\nabla_\theta \left\|\nabla_\phi f(w^{(k)})\right\|^2$$

where the generator updates remain the same with SimGD, and thus the corresponding regularization matrix is $\Gamma = \begin{bmatrix} I & 0 \\ \gamma\nabla^2_{\theta\phi}f(w^{(k)}) & I \end{bmatrix}$.

Their convergence behaviors in terms of stabilizing the simple vanilla GAN example (3) are given as follows.

**Theorem 2.** *In the simple vanilla GAN example, none of the previous gradient-based regularization methods (i.e., only regularizing generator, ConOpt and only regularizing discriminator) are capable of simultaneously alleviating the Phase Factor and Conditioning Factor.*

*Proof:* See Appendix D.1. □

From the above theorem, together with the example of changing the representations in Appendix B, we can see that without carefully taking into account both the *Phase Factor* and *Conditioning Factor*, these GAN variants might still suffer from the poor convergence even in the simple GAN example.

### 5.2 JARE

Based on the above theoretical analysis, we propose a new but simple Jacobian regularization, called JARE, which also applies the regularization terms based on the

gradients of the generator and discriminator. Specifically, the regularized updates are given by

$$\phi^{(k+1)} = \phi^{(k)} - \eta\nabla_\phi f(w^{(k)}) - \frac{1}{2}\eta\gamma\nabla_\phi \left\|\nabla_\theta f(w^{(k)})\right\|^2$$
$$\theta^{(k+1)} = \theta^{(k)} + \eta\nabla_\theta f(w^{(k)}) - \frac{1}{2}\eta\gamma\nabla_\theta \left\|\nabla_\phi f(w^{(k)})\right\|^2$$
$$(7)$$

Similarly, the corresponding regularization matrix is $\Gamma = \begin{bmatrix} I & -\gamma\nabla^2_{\phi\theta}f(w^{(k)}) \\ \gamma\nabla^2_{\theta\phi}f(w^{(k)}) & I \end{bmatrix}$ with $\gamma > 0$ being a tunable hyperparameter.

Note that the key difference between JARE and ConOpt is that JARE does not introduce the Hessians $\nabla^2_{\phi\phi}f(w^{(k)})$ and $\nabla^2_{\theta\theta}f(w^{(k)})$ in the regularization matrix $\Gamma$. Intuitively, a reason for not doing this is to avoid the risk of reversing the gradient flows, which may diverge the GAN training dynamics (see Appendix C for a detailed explanation). The following theorem shows the eigenvalues of the Jacobian in the simple vanilla GAN example trained via the proposed method.

**Theorem 3.** *For any point within $B_\delta(w^*)$, the Jacobian $A$ in the simple vanilla GAN example trained via JARE has the following eigenvalues: $\lambda_{1,2}(A) = \frac{-(\sigma^2+\gamma)\pm\sqrt{(\sigma^2+\gamma)^2-(\gamma^2+4)}}{4}$ and $\lambda_{3,4}(A) = \frac{-(\beta^2+\gamma)\pm\sqrt{(\beta^2+\gamma)^2-(\gamma^2+4)}}{4}$, where $\beta^2 \triangleq \sigma^2 + \|v\|^2$.*

*Proof:* See Appendix D.2. □

From the above theorem, given $0 < \sigma^2 < 2$ and $\beta^2 > 2$, we can evaluate both $\zeta \triangleq \left|\frac{\text{Im}\{\lambda_{1,2}(A)\}}{\text{Re}\{\lambda_{1,2}(A)\}}\right|$ and $\tau \triangleq \left|\frac{\lambda_3(A)}{\lambda_4(A)}\right|$, two key variables that reflect the impact of the *Phase Factor* and *Conditioning Factor*, respectively, and see how the tunable parameter $\gamma$ in JARE changes their values. The results are given in the following corollary.

**Corollary 3.** *In the simple vanilla GAN example trained via JARE, $\zeta$ monotonically decreases as $\gamma$ increases, and if $\gamma \geq 2$, $\tau$ also monotonically decreases as $\gamma$ increases. In the limit of $\gamma \to \infty$, we get $\zeta \to 0$ (i.e., no complex eigenvalues) and $\tau \to 1$ (i.e., well conditioned). Therefore, we can make $\gamma$ large enough in JARE to alleviate the impact of the Phase Factor and Conditioning Factor simultaneously.*

*Proof.* See Appendix D.3. □

As we know from Corollary 1 and 2, if $\zeta \to 0$ and $\tau \to 1$, the non-asymptotic convergence rate will be increasingly improved. Therefore, the above corollary justifies that the proposed JARE will provide a good local convergence behavior by applying a reasonably large hyperparameter $\gamma$. However, we cannot make $\gamma$ arbitrarily large in JARE. According to the non-asymptotic analysis, the

step size in JARE should satisfy $0 < \eta < \eta_{\max}$ where $\eta_{\max} \triangleq 8 \min\{\frac{\gamma+\sigma^2}{\gamma^2+4}, \frac{1}{(\beta^2+\gamma)+\sqrt{2\beta^2\gamma+(\beta^2)^2-4}}\}$. As we can see $\eta_{\max}$ decreases with the increment of $\gamma$, and goes to 0 as $\gamma \to \infty$. So when $\gamma$ is sufficiently large, we have to make the step size infinitely small accordingly.

## 6 EXTENSIONS TO GENERAL GANS

The above analysis is based on the simple GAN example, and here we can extend it to the more general GAN scenario. First, we show that the two factors identified above can also be significant issues in general GANs. Second, we show that JARE can alleviate both of these factors simultaneously in general GANs.

For ease of analysis, we make the following assumption in terms of equilibrium point in general GANs (Nagarajan and Kolter, 2017; Mescheder et al., 2018).

**Assumption 1.** *In equilibrium, the optimal generated distribution satisfies $p_{\phi^*} = p_r$, and the optimal discriminator satisfies $D_{\theta^*}(x) = 0$ for the local neighborhood of any $x \in \mathcal{X}$.*

Basically, this assumption means that the generator is powerful enough to match the true data distribution in equilibrium where the discriminator cannot distinguish the real and fake samples. In order to avoid trivial solutions, we also make an assumption as follows (Nagarajan and Kolter, 2017).

**Assumption 2.** *The two concave functions $g_1$ and $g_2$ satisfy $g_1''(0) + g_2''(0) < 0$ and $g_1'(0) = -g_2'(0) \neq 0$.*

For example, the vanilla GAN loss and reverse KL divergence satisfy this assumption but the WGAN loss does not. Under these two assumptions, the Jacobian of general GANs is given as follows:

**Lemma 3.** *For an equilibrium point $(\phi^*, \theta^*)$ satisfying Assumptions 1 and 2, the Jacobian $A$ in general GANs trained via SimGD can be written in the form*

$$A = \begin{bmatrix} 0 & -P \\ P^T & Q \end{bmatrix} \tag{8}$$

*where $P \in \mathbb{R}^{m \times n}$ and $Q \in \mathbb{R}^{m \times m}$ are given by*

$$\begin{aligned} P &= g_2'(0)\mathbb{E}_{z \sim P_0}[\nabla_\phi G_\phi(z)\nabla_{x\theta}^2 D_\theta(x)]|_{x=G_\phi(z)} \\ Q &= (g_1''(0) + g_2''(0))\mathbb{E}_{x \sim P_r}[\nabla_\theta D_\theta(x)\nabla_\theta D_\theta(x)^T] \end{aligned} \tag{9}$$

*Proof:* See Appendix E.1. □

The off-diagonal matrix $P$ represents how sensitive the discriminator is to the generator's local updates. The diagonal matrix $Q$ represents the local geometry of the discriminator. It is easy to verify that the Jacobian $A$ in

the simple GAN example is a special case of Lemma 3. Note that for WGAN, since $g_1''(0) = g_2''(0) = 0$, we have $Q = 0$ and therefore it is not even asymptotically convergent (Nagarajan and Kolter, 2017), rendering a convergence rate analysis irrelevant. We are now ready to compute the eigenvalues of the Jacobian $A$ in general GANs.

**Theorem 4.** *For the equilibrium point $(\phi^*, \theta^*)$ satisfying Assumptions 1 and 2, the eigenvalues of the Jacobian $A$ in general GANs trained via SimGD can be written in the form*

$$\lambda(A) = \frac{a_1 \pm \sqrt{a_1^2 - 4a_2}}{2} \tag{10}$$

*where $a_1$ and $a_2$ are certain convex combinations of the eigenvalues of $Q$ and $P^T P$, respectively. That is,*

$$a_1 = \sum_{i=1}^{m} \alpha_i \lambda_i(Q), \quad a_2 = \sum_{i=1}^{m} \tilde{\alpha}_i \lambda_i(P^T P) \tag{11}$$

*for some coefficients $\alpha_i \geq 0$ with $\sum_{i=1}^{m} \alpha_i = 1$ and some coefficients $\tilde{\alpha}_i \geq 0$ with $\sum_{i=1}^{m} \tilde{\alpha}_i = 1$.*

*Proof:* See Appendix E.2. □

Let $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimum and maximum eigenvalues of a square matrix, respectively. From Lemma 3 we know $Q \preceq 0$, so $\lambda_{\min}(Q) \leq \lambda_i(Q) \leq \lambda_{\max}(Q) \leq 0, \forall i$. Also, by definition we have $P^T P \succeq 0$, so $0 \leq \lambda_{\min}(P^T P) \leq \lambda_i(P^T P) \leq \lambda_{\max}(P^T P), \forall i$. The convex combination in Eq. (11) then implies

$$\lambda_{\min}(Q) \leq a_1 \leq \lambda_{\max}(Q) \leq 0$$
$$0 \leq \lambda_{\min}(P^T P) \leq a_2 \leq \lambda_{\max}(P^T P)$$

Therefore, how to balance the eigenvalue distributions of $Q$ and $P^T P$ plays an essential role in determining the eigenvalues of the Jacobian $A$. To see this, we consider two relatively extreme cases as follows:

First, if the maximum absolute value of the eigenvalues of $Q$ is much smaller than the minimum absolute value of the eigenvalues of $P^T P$, in particular we assume $Q$ and $P^T P$ satisfy $c|\lambda_{\min}(Q)|^2 = 4|\lambda_{\min}(P^T P)|$ with $c \gg 1$, then for any coefficients $\alpha_i$ and $\tilde{\alpha}_i$ in Theorem 4, we have $a_1^2 < 4a_2$ and thus $\lambda(A)$ is complex-valued with the imaginary-to-real ratio satisfying $\zeta = \sqrt{c-1}$. We can see that as $c$ becomes larger, the impact of *Phase Factor* will be more severe.

Second, if the minimum absolute value of the eigenvalues of $Q$ is much larger than the maximum absolute value of the eigenvalues of $P^T P$, in particular we assume $Q$ and $P^T P$ in some GAN scenario satisfy $c'|\lambda_{\max}(Q)|^2 = 4|\lambda_{\max}(P^T P)|$ with $c' \ll 1$, then for any coefficients $\alpha_i$ and $\tilde{\alpha}_i$ in Theorem 4, we have $a_1^2 > 4a_2$ and thus $\lambda(A)$

| Requirements | stable SimGD | stable JARE |
|---|---|---|
| $Q$ is well-conditioned | ✓ | |
| $P^T P$ is well-conditioned | ✓ | ✓ |
| $Q$ matches $P^T P$ | ✓ | |

Table 1: The general requirements of ensuring a good local convergence behavior in both the GAN trained via SimGD (called "stable SimGD") and the GAN trained via the proposed JARE (called "stable JARE"). The more requirements that the GAN training needs, the more difficult to ensure a good local convergence behavior.

is real-valued with the absolute value of eigenvalue ratio satisfying $\tau = (\sqrt{\frac{1}{c'}} + \sqrt{\frac{1}{c'} - 1})^2$. We can observe that when $c'$ is smaller, the impact of *Conditioning Factor* will also be increasingly severe.

Therefore, even if $Q$ and $P^T P$ themselves are both well-conditioned with all real eigenvalues, there still exist either the *Phase Factor* or *Conditioning Factor* in the Jacobian $A$ due to the imbalance between their eigenvalue distributions. More generally in real GANs, it is also likely that either $Q$ or $P^T P$ becomes ill-conditioned, the GAN training dynamics will suffer more from the co-existence of *Phase Factor* and the *Conditioning Factor*. Note that the simple vanilla GAN example in Section 4 is just an illustrative special case of general GANs.

In summary, we need to make sure that $Q$ and $P^T P$ are both well-conditioned (which requires a well-designed generator and discriminator) *and* have similar eigenvalues (which requires the discriminator to well match the generator) to avoid these two factors in general GANs. Generally, these requirements are difficult to satisfy, which explains why GANs are hard to train and also why they are so sensitive to network architectures and other hyperparameters.

Next, we compute the eigenvalues of the Jacobian $A$ in general GANs trained with our proposed JARE.

**Theorem 5.** *For the equilibrium point $(\phi^*, \theta^*)$ satisfying Assumptions 1 and 2, the eigenvalues of the Jacobian $A$ in general GANs trained via JARE satisfy that in the limit $\gamma \to \infty$,*

$$\lambda(A) = -\gamma \lambda(P^T P) \tag{12}$$

*Proof:* See Appendix E.3. □

We can see that $\lambda(A)$ in (12) is real-valued, which means there is no *Phase Factor* any more with a sufficiently large regularization term $\gamma$ in JARE. Also, the eigenvalue distribution of the Jacobian $A$ now *only* depends on $P^T P$, which means in general GANs trained via JARE, the imbalance between the eigenvalue distributions of $Q$
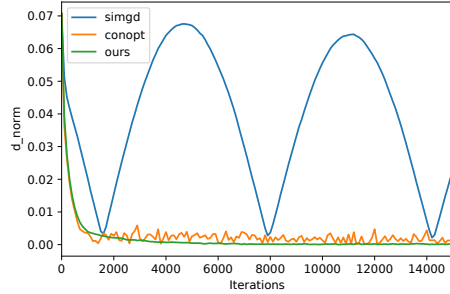


(a) Discriminator training curve
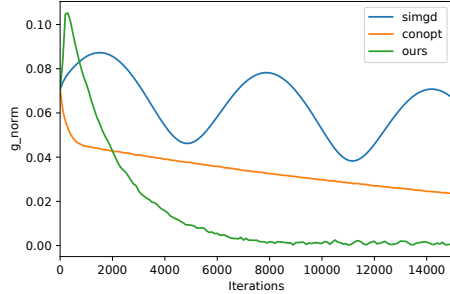


(b) Generator training curve

Figure 1: Training dynamics of SimGD, ConOpt and JARE (Ours) in the simple vanilla GAN example where $\mu = 4$ and $\sigma^2 = 0.04$. (a) shows the discriminator convergence where "d_norm" denotes the $l_2$ distance between current and optimal value of the discriminator parameters, and (b) shows the generator convergence where "g_norm" denotes the $l_2$ distance between current and optimal value of the generator parameters.

and $P^T P$ will not result in an undesirable properties of the Jacobian $A$. Instead, we only need to make sure $P^T P$ is well-conditioned to achieve a good convergence behavior. The comparison between GANs trained via SimGD and JARE regarding the requirements of good training dynamics is illustrated in Table 1. In this sense, JARE will be significantly easier to train, with greater stability and more robustness to different network architectures and hyperparameters.

## 7 EXPERIMENTS

**Isotropic Gaussian.** First, we empirically verify our theory in the simple vanilla GAN example. Specifically, we consider a two-dimensional case, i.e. $n = 2$ and the mean of true data is $v = [0, \mu]^T$. To test the local convergence, the parameters of both the discriminator and generator are initialized within $B_\delta(w^*)$ where $\delta = 0.05$. For hyperparameters, we set the learning rate to be $\eta = 0.001$, the regularization parameter to be $\gamma = 10$, the optimizer to be stochastic gradient descent (SGD) with a batch size 128, and run 15K iterations.

Figure 1 shows the discriminator and generator training curves, respectively, for three training methods: SimGD,

(a) SimGD



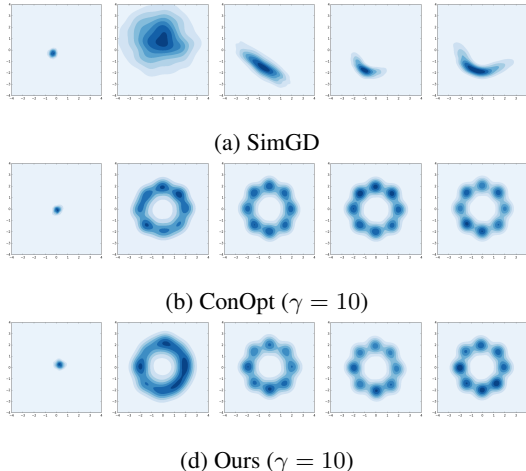(b) ConOpt ($\gamma = 10$)



(d) Ours ($\gamma = 10$)

Figure 2: Comparison of SimGD, ConOpt and JARE (Ours) on the mixture of Gaussians over iterations where $r = 2$ and $\gamma = 10$ for both ConOpt and JARE. From left to right, each row consists of the results after 0, 2000, 4000, 6000, 8000 iterations.
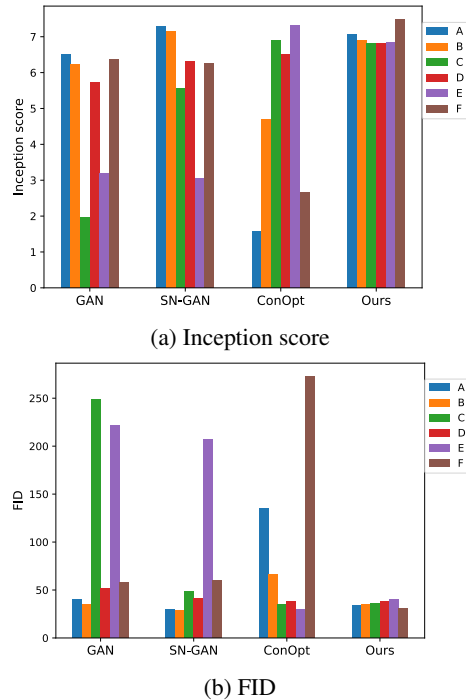


(a) Inception score



(b) FID

Figure 3: Inception scores and FIDs of different methods: GAN, SN-GAN, ConOpt and JARE (Ours) in different GAN settings (A-F) on CIFAR-10. For inception score, the higher is better and for FID, the lower is better.

ConOpt and JARE (Ours) by letting $\mu = 4$ and $\sigma^2 = 0.04$. We observe that for SimGD, the training curves oscillate with very weak damping which becomes even weaker if we increase $\mu$ or decrease $\sigma^2$ (See Figures 5 and 6 in Appendix F.1). It verifies that SimGD suffers from poor convergence caused by the *Phase Factor* and *Conditioning Factor*. Also, ConOpt could alleviate the *Phase Factor* since oscillations caused by complex eigenvalues disappear. However, its generator convergence is heavily slowed down by the *Conditioning Factor* which becomes worse as we increase $\mu$ (See Figure 5 in Appendix F.1). In contrast, the JARE enjoys a decent convergence rate for both the generator and discriminator by alleviating the two factors simultaneously.

**Mixture of Gaussians.** We also test JARE in a commonly used toy example where the goal is to learn a mixture of Gaussians with modes uniformly distributed around a circle with radius $r$. Here we set $r = 2$ while keeping other settings and network architectures the same with Mescheder et al. (2017). We run SimGD, ConOpt and JARE (Ours) with RMSProp (Tieleman and Hinton, 2012) and learning rate of $10^{-4}$ for 10K iterations, and the input noise is sampled from a 64-dimensional Gaussian $\mathcal{N}(0, rI_{64})$. Figure 2 shows their results over different iterations. We can see that SimGD oscillates among different modes and fails to converge, while ConOpt and JARE can both converge to the target data distribution. Please see Figures 7 and 8 in Appendix F.2 for more detailed comparisons among these methods, where we show JARE with $\gamma = 1000$ tends to behave slightly better than ConOpt in more difficult settings.

**CIFAR-10.** In this experiment, we quantitatively evalu-

ate the sample quality of JARE on the CIFAR-10 dataset (Torralba et al., 2008) with the *inception score* (Salimans et al., 2016) and *Frechet inception distance (FID)* (Heusel et al., 2017). We test the dependencies of JARE on different network architectures and hyperparameters. We compare with other GAN training methods, including the standard GAN (Goodfellow et al., 2014) (denoted as 'GAN'), ConOpt (Mescheder et al., 2017) and SN-GAN (Miyato et al., 2018). For all methods, we use the non-saturating loss as suggested in Goodfellow et al. (2014). For fair comparison, we test 6 settings: the standard CNN model in Miyato et al. (2018) with batch normalization (Ioffe and Szegedy, 2015) on generator (A) or without batch normalization on generator (B), the DCGAN-like architecture with a constant number of filters in Mescheder et al. (2017) via the Adam optimizer (Kingma and Ba, 2015) (C) or via the RMSProp optimizer (D), and the ResNet (He et al., 2016) architectures v1 (E) or v2 (F) with a constant number of filters where $M_f = 64$. Please see Appendix F.3 more details. Unless otherwise stated, we use the Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. Also, we use a batch size of 64 and run all experiments with a learning rate of $10^{-4}$ for 500K iterations. For ConOpt, we set $\gamma = 10$, and for JARE, we set $\gamma = 100$.
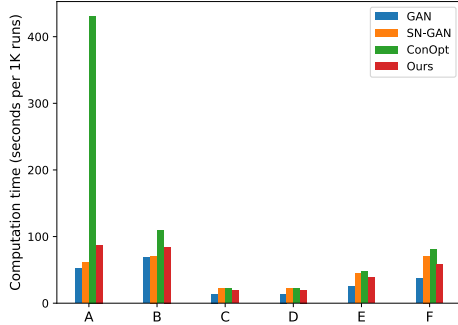
Figure 4: Training time on CIFAR-10 with different training methods: GAN, SN-GAN, ConOpt and JARE (Ours) in all the A-F settings.

Figure 3 shows the inception scores and FIDs for different training methods with all 6 settings on CIFAR-10 (also see Figure 9 in Appendix F.4 for the generated samples). We can see that JARE is more robust than other methods regarding different network architectures and tuning hyperparameters, which shows the potential advantages of JARE in stabilizing the real GAN training. Both SN-GAN and ConOpt perform almost the best in their own proposed GAN architectures but perform poorly in other cases. Besides, the training time (in seconds per 1K runs) on CIFAR-10 with these different methods is given in Table 4. We can see that the training time of JARE is always lower than ConOpt and on par with SN-GAN, which is not much higher than the standard GAN, which means the extra computational cost introduced by the regularization terms in JARE is relatively low at least in there settings.

## 8   DISCUSSION AND CONCLUSIONS

In this paper, we first analyzed the non-asymptotic local convergence behavior of GAN training dynamics in a simple GAN example and later extended the analysis to the general GAN scenario. We found out that in order to ensure a good convergence behavior in GANs, both the *Phase Factor* and *Conditioning Factor* need to be addressed simultaneously. However, we showed that previous gradient-based regularizations can only avoid one factor while making the other more severe. Therefore, we proposed a new Jacobian regularization for GANs, called JARE, and showed theoretically it can alleviate the two factors simultaneously. Finally, we did experiments on isotropic Gaussian, mixture of Gaussians and CIFAR-10 to show the training stability of JARE.

However, the proposed JARE also has its limitation: Although it is constructed in a principled way and the preliminary experimental results showed its potential benefits at stabilizing GANs, in order to scale to the large-scale GAN training (Brock et al., 2019), we will need to get rid of the computationally expensive second-order derivatives in the regularization term of JARE. A potential direction of alleviating this limitation could be interpreting JARE as a form of *adversarial extrapolation* where two agents playing the game anticipate each other's learning updates, which we think is game-theoretic, intuitive and thought-provoking.

For example, different from current extrapolation methods applied in GANs (Gidel et al., 2019; Daskalakis et al., 2018; Yadav et al., 2018), we can introduce the *adversarial extrapolation* as follows:

$$
\begin{aligned}
\phi^{(k+1)} &= \phi^{(k)} - \eta \nabla_\phi f(\phi^{(k)}, \theta^{(k+\frac{1}{2})}) \\
\theta^{(k+1)} &= \theta^{(k)} + \eta \nabla_\theta f(\phi^{(k+\frac{1}{2})}, \theta^{(k)})
\end{aligned}
\tag{13}
$$

where the intermediate prediction terms $\phi^{(k+\frac{1}{2})}$ and $\theta^{(k+\frac{1}{2})}$ are given by

$$
\begin{aligned}
\phi^{(k+\frac{1}{2})} &= \phi^{(k)} - \frac{\gamma}{2} \nabla_\phi f(\phi^{(k)}, \theta^{(k)}) \\
\theta^{(k+\frac{1}{2})} &= \theta^{(k)} + \frac{\gamma}{2} \nabla_\theta f(\phi^{(k)}, \theta^{(k)})
\end{aligned}
\tag{14}
$$

As we can see, by applying the first-order Taylor approximation with respect to $\theta^{(k+\frac{1}{2})}$ and $\phi^{(k+\frac{1}{2})}$, respectively, we have

$$
f(\phi^{(k)}, \theta^{(k+\frac{1}{2})}) \approx f(\phi^{(k)}, \theta^{(k)}) + \frac{\gamma}{2} \left\| \nabla_\theta f(\phi^{(k)}, \theta^{(k)}) \right\|^2
$$

$$
f(\phi^{(k+\frac{1}{2})}, \theta^{(k)}) \approx f(\phi^{(k)}, \theta^{(k)}) - \frac{\gamma}{2} \left\| \nabla_\phi f(\phi^{(k)}, \theta^{(k)}) \right\|^2
$$

Thus, by substituting the above two approximations into (13), we can get the proposed JARE in (7).

From the above analysis, the new adversarial extrapolation method derived directly from JARE may enjoy both our theoretical grounding and the computational benefits of the first-order optimization methods. We leave the further investigation of the adversarial extrapolation method, and the relationship between JARE and other extrapolation methods in GANs, as the future work.

## Acknowledgement

## References

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

Arrowsmith, D. and Place, C. M. (1992). *Dynamical systems: differential equations, maps, and chaotic behaviour*, volume 5. CRC Press.

Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., and Graepel, T. (2018). The mechanics of n-player differentiable games. In *International Conference on Machine Learning (ICML)*, pages 363–372.

Brock, A., Donahue, J., and Simonyan, K. (2019). Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*.

Daskalakis, C., Ilyas, A., Syrgkanis, V., and Zeng, H. (2018). Training gans with optimism. In *International Conference on Learning Representations (ICLR)*.

Daskalakis, C. and Panageas, I. (2018). The limit points of (optimistic) gradient descent in min-max optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9256–9266.

Gidel, G., Berard, H., Vincent, P., and Lacoste-Julien, S. (2019). A variational inequality perspective on generative adversarial nets. In *International Conference on Learning Representations (ICLR)*.

Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems (NeurIPS)*, pages 2672–2680.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5769–5779.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6629–6640.

Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4565–4573.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456.

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

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Letcher, A., Foerster, J., Balduzzi, D., Rocktäschel, T., and Whiteson, S. (2019). Stable opponent shaping in differentiable games. In *International Conference on Learning Representations (ICLR)*.

Li, J., Madry, A., Peebles, J., and Schmidt, L. (2018). On the limitations of first-order approximation in gan dynamics. In *Proceedings of International Conference on Machine Learning (ICML)*.

Liang, T. and Stokes, J. (2019). Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Mescheder, L., Geiger, A., and Nowozin, S. (2018). Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, pages 3478–3487.

Mescheder, L., Nowozin, S., and Geiger, A. (2017). The numerics of gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1823–1833.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*.

Nagarajan, V. and Kolter, J. Z. (2017). Gradient descent gan optimization is locally stable. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5591–5600.

Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 271–279.

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

Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. (2017). Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2015–2025.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2234–2242.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5rmsprop: Divide the gradient by a running average of its recent magnitude.

Torralba, A., Fergus, R., and Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970.

Yadav, A., Shah, S., Xu, Z., Jacobs, D., and Goldstein, T. (2018). Stabilizing adversarial nets with prediction methods. In *International Conference on Learning Representations (ICLR)*.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2223–2232.