Understanding and Stabilizing GANs' Training Dynamics using Control Theory

Kun Xu [1] **Chongxuan Li** [1] **Jun Zhu** [1] **Bo Zhang** [1]

Abstract

Generative adversarial networks (GANs) are effective in generating realistic images but the training is often unstable. There are existing efforts that model the training dynamics of GANs in the parameter space but the analysis cannot directly motivate practically effective stabilizing methods. To this end, we present a conceptually novel perspective from control theory to directly model the dynamics of GANs in the function space and provide simple yet effective methods to stabilize GANs' training. We first analyze the training dynamic of a prototypical Dirac GAN and adopt the widely-used closed-loop control (CLC) to improve its stability. We then extend CLC to stabilize the training dynamic of normal GANs, where CLC is implemented as a squared $L2$ regularizer on the output of the discriminator. Empirical results show that our method can effectively stabilize the training and obtain state-of-the-art performance on data generation tasks.

1. Introduction

Generative adversarial networks (GANs) (Goodfellow et al., 2014) have shown promise in generating realistic natural images (Brock et al., 2018) and facilitating unsupervised and semi-supervised learning (Chen et al., 2016; Li et al., 2017; Donahue & Simonyan, 2019). In GANs, an implicit generator $\mathcal{G}$ is defined by mapping a noise distribution to the data space. Since no density function is defined for the implicit generator, the maximum likelihood estimate is infeasible for GANs. Instead, a discriminator $\mathcal{D}$ is introduced to estimate the density ratio between the data distribution p and the generating distribution $p_{\mathcal{G}}$ by telling the real samples from fake ones. $\mathcal{G}$ aims to recover the data distribution by maximizing this ratio. This framework is formulated as a minimax optimization problem, which can be solved by optimizing $\mathcal{G}$ and $\mathcal{D}$ alternately. In practice, however, GANs suffers from the instability of training (Goodfellow, 2016), where divergency and oscillations are often observed (Liang et al., 2018; Chavdarova & Fleuret, 2018).

Early methods (Mao et al., 2017; Gulrajani et al., 2017; Arjovsky et al., 2017; Du et al., 2018) introduce different types of divergences to improve the training process of GANs. Their theoretical analyses assume that $\mathcal{D}$ achieves its optimum when training $\mathcal{G}$. However, the practical training process (e.g., alternative stochastic gradient descent) often violates the above assumption and therefore is not guaranteed to converge to the desired equilibrium. Several empirical regularizations (Miyato et al., 2018; Gulrajani et al., 2017; Zhang et al., 2019) are used to improve the training process whereas no stability can be guaranteed.

Recently, Mescheder et al. (2017) and Nagarajan & Kolter (2017) directly model the training dynamics of GANs, i.e. how the parameters develop over time. Formally, the dynamic is defined as the gradient flow of the parameters. The stability of the dynamic is fully determined by the eigenvalues of the Jacobian matrix of the gradient flow. Indeed, the stability analysis in a linear prototypical GAN (i.e. Dirac GAN (Mescheder et al., 2018)) is elegant. However, this analysis does not directly motivate effective algorithms to stabilize GANs' training. To our knowledge, such methods do not report competitive image generation results to the state-of-the-art GANs (Miyato et al., 2018).

In this paper, we understand and stabilize GANs' training dynamics from the perspective of control theory. Based on the recipe for control theory, we can not only analyze the dynamics of Dirac GAN formally, but also develop practically effective stabilizing methods for nonlinear dynamics (Khalil, 2002). Specifically, we start from revisiting the Dirac GAN example with the WGAN's objective function in Sec. 3. By utilizing the Laplace transform (Widder, 2015) (LT), the training dynamics of both $\mathcal{D}$ and $\mathcal{G}$ can be modeled in the *frequency domain* instead of the *time domain* in previous methods (Mescheder et al., 2017; 2018). These types of dynamics are well studied in control theory and

[1]Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center, Tsinghua-Bosch ML Center, THBI Lab, Tsinghua University, Beijing, China. Correspondence to: Jun Zhu <dcszj@mail.tsinghua.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

the stability can be easily inferred. The analysis can be simply generalized to other objective functions with *local linearization*. Given the instability of GANs, the recipe for control theory provides a set of tools to stabilize their dynamics. We first adopt the *closed-loop control* (CLC) to successfully stabilize the dynamic of Dirac GAN with theoretical guarantee. Besides, extensive empirical results in control theory show that the CLC is also helpful in non-linear settings (Khalil, 2002). It inspires us to extend our proposal to normal GANs by modeling $\mathcal{D}$ and $\mathcal{G}$'s dynamics in the function space where these dynamics and Dirac GAN's dynamics share similar forms and characters. The CLC is implemented as a regularization term to $\mathcal{D}$'s objective function which penalizes the squared $L2$ norm of the output of $\mathcal{D}$ as we described in Sec. 4.1. We therefore refer our method as CLC-GAN. CLC-GAN is verified on an 1-dimension toy example as well as the natural images including CIFAR10 (Krizhevsky et al., 2009) and CelebA (Liu et al., 2015). The results demonstrate that our method can successfully stabilize the dynamics of GANs and achieve state-of-the-art performance.

Our contributions are summarized as:

- We formally analyze the training dynamics of GANs from a novel perspective of control theory, which is generally applicable to different objective functions.
- We propose to use the CLC as an effective method to stabilize the training of GANs, while other advanced control methods can be explored in future.
- The simulated results on Dirac GAN agree with the theoretical analysis and CLC-GAN achieves the state-of-the-art performance on natural image generations.

2. Preliminary

In this section, we present the recipe for control theory, especially under the Laplace transform, which is powerful to model dynamic systems and design stabilizing methods.

2.1. Modeling Dynamic Systems

In control theory, a *signal* is represented as a function over time t, i.e., in the *time domain* (Kailath, 1980). A dynamic[1] represents how one signal (i.e., output, denoted by $\boldsymbol{y}(t)$) develops with respect to another signal (i.e., input, denoted by $\boldsymbol{u}(t)$) over time. A natural representation of a dynamic is a differential equation (DE)[2]:

$$\frac{d\boldsymbol{y}(t)}{dt} = f(\boldsymbol{y}(t), \boldsymbol{u}(t)), \tag{1}$$

together with an initial condition $\boldsymbol{y}(0) = \boldsymbol{y}_0$. Note that $f(\cdot,\cdot)$, $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$ can be vector valued functions. We assume $\boldsymbol{y}_0 = 0$ unless specified. A dynamic is *linear* if $f(\cdot,\cdot)$ is a linear function.

Besides the time domain, a signal can also be represented as a function of frequency s, i.e., in the *frequency domain*. A DE of a linear dynamic in the time domain can be converted to a simple algebraic equation in the frequency domain, which can largely simplify the solving process and stability analysis of a dynamic. Laplace transform (Widder, 2015) (LT) is a widely-adopted operator to convert signals from the time domain to the frequency domain. Formally, LT is given by:

$$\mathcal{F}(\boldsymbol{h})(s) = \int_0^\infty \boldsymbol{h}(t)e^{-st}dt = \boldsymbol{H}(s), \tag{2}$$

where $\boldsymbol{h}$ is a signal in the time domain, and $s = \sigma + \omega i \in \mathbb{C}$ with real numbers σ and ω. The real and imaginary parts of $\boldsymbol{H}(s) \in \mathbb{C}$ denote the gain and phase of the frequency s in $\boldsymbol{h}$. In this paper, we use bold lowercase letters (e.g., $\boldsymbol{y}, \boldsymbol{u}$) to denote signals in the time domain and bold capital letters (e.g. $\boldsymbol{Y}, \boldsymbol{U}$) to denote signals in the frequency domain.

Leveraging LT, the derivation over time t can be represented as multiplying a factor s in the frequency domain:

$$\mathcal{F}(\frac{d\boldsymbol{h}(t)}{dt}) = s\mathcal{F}(\boldsymbol{h}). \tag{3}$$

Therefore, by applying LT to both sides of a DE in Eqn. (1), a linear dynamic can be solved by the formal rules of algebra and represented in the form of $\boldsymbol{Y}(s) = \boldsymbol{T}(s)\boldsymbol{U}(s)$, where $\boldsymbol{T}(s)$ is a simple rational fraction called *transfer function* (Kailath, 1980). The transfer function can facilitate the stability analysis, as detailed in Sec. 2.2.

2.2. Stability Analysis

In general, we require a dynamic to be *stable*. Although different definitions exist, we consider the widely adopted asymptotic stability[3] (Kailath, 1980) in this paper.

Definition 1. *For a constant input $\boldsymbol{u}(t) = \boldsymbol{u}_c$, a point $\boldsymbol{y}_e$ is called an equilibrium point of a dynamic represented in Eqn. (1), if $f(\boldsymbol{y}_e, \boldsymbol{u}_c) = 0$. A dynamic is called asymptotically stable if for every $\epsilon > 0$, there exists $\sigma > 0$ such that if $||\boldsymbol{y}(0) - \boldsymbol{y}_e|| < \sigma$, then for every $t > 0$, $||\boldsymbol{y}(t) - \boldsymbol{y}_e|| < \epsilon$ and $\lim_{t\to\infty} ||\boldsymbol{y}(t) - \boldsymbol{y}_e|| = 0$. Here $||\cdot||$ is a norm defined in the vector space of $\boldsymbol{y}$.*

In the frequency domain, the stability can be directly inferred from the transfer function. Formally, we define *poles* as the roots of the denominator in a transfer function. The stability of a linear dynamic is fully determined by its poles as summarized in the following proposition.

[1] For simplicity, we use *dynamic* for dynamic system.

[2] We consider ordinary differential equations in this paper.

[3] This definition is consistent with existing work in Mescheder et al. (2017) and Mescheder et al. (2018).

Proposition 1. *(Theorm 2.6-1 in Kailath (1980))*

1. *A dynamic is asymptotic stable if all poles have negative real parts.*
2. *A dynamic is oscillatory (i.e., bounded output but not stable) if one or more poles are purely imaginary.*
3. *A dynamic is diverged (unbounded output) if one or more poles have positive real parts.*

2.3. Control Methods

For an unstable dynamic, control theory provides a set of methods to improve its stability. Among them, the *closed-loop control* (Kailath, 1980) (CLC) is one of the most popular ones and robust to nonlinearity in dynamics practically.

The central idea is to modify the transfer function by feeding the output back to the input such that all poles have negative real parts. Specifically, we introduce an additional dynamics called *controllers* with transfer functions $\boldsymbol{T}_b(s)$ to adjust the output signal and input signal respectively. The controller takes $\boldsymbol{Y}(s)$ as input and output the feedback signal $\boldsymbol{Y}_b = \boldsymbol{T}_b(s)\boldsymbol{Y}(s)$. We then substitute the difference between $\boldsymbol{U}$ and $\boldsymbol{Y}_b$ (i.e., $\boldsymbol{M} = \boldsymbol{U} - \boldsymbol{Y}_b$) for input in the original dynamics, resulting the output signal as $\boldsymbol{Y}(s) = \boldsymbol{T}(s)\boldsymbol{M}(s)$. The relationship between the input $\boldsymbol{U}(s)$ and the output $\boldsymbol{Y}(s)$ is:

$$\boldsymbol{Y}(s) = \boldsymbol{T}(s)(\boldsymbol{U}(s) - \boldsymbol{T}_b(s)Y(s)). \tag{4}$$

Further, the whole controlled dynamic is given as:

$$\boldsymbol{Y}(s) = \frac{\boldsymbol{T}(s)}{1 + \boldsymbol{T}_b(s)\boldsymbol{T}(s)}\boldsymbol{U}(s). \tag{5}$$

With a properly designed $\boldsymbol{T}_b$, the poles of the dynamic in Eqn. (5) can have negative real parts and the dynamic is stabilized. In the following, we first model and stabilize the training dynamic of Dirac GAN: a simplified GAN with linear dynamics in Sec. 3 and then we generalize it to the realistic setting in Sec. 4.

3. Analyzing Dirac GAN by Control Theory

In this section, we focus on the Dirac GAN (Mescheder et al., 2018), which is a widely adopted example to analyze the stability of GANs. Previous work (Mescheder et al., 2017; Gidel et al., 2018) uses the Jacobian matrix to analyze the stability of dynamics whereas does not directly provide an approach to stabilize it. Instead, we revisit this example from the perspective of control theory and develop a principled method that not only analyzes but also improves the stability of various GANs.

3.1. Modeling Dynamics

We first model the dynamics of the Dirac GANs in the language of control theory, which can facilitate the stability analysis and improvement in Sec. 3.2. In Dirac GAN, $\mathcal{G}$ is defined as $p_{\mathcal{G}}(x) = \delta(x - \theta)$ where $\delta(\cdot)$ is the Dirac delta function, and $\mathcal{D}$ is defined as $\mathcal{D}(x) = \phi x$. θ and ϕ are the parameters of $\mathcal{G}$ and $\mathcal{D}$ respectively. The data distribution is $p(x) = \delta(x - c)$ with a constant c. Generally, the objective functions of $\mathcal{D}$ and $\mathcal{G}$ can be written as:

$$\begin{aligned} \max_{\phi} \mathcal{V}_1(\phi; \theta) &= h_1(\mathcal{D}(c)) + h_2(\mathcal{D}(\theta)), \\ \max_{\theta} \mathcal{V}_2(\theta; \phi) &= h_3(\mathcal{D}(\theta)). \end{aligned} \tag{6}$$

Here $h_i(\cdot) : \mathbb{R} \to \mathbb{R}$ is a scalar function for $i \in \{1, 2, 3\}$. Assuming that the equilibrium point of $\mathcal{D}$ is a zero function as in most GANs (Goodfellow et al., 2014; Arjovsky et al., 2017), it is required that $h_1(\cdot)$ and $h_3(\cdot)$ are increasing functions and $h_2(\cdot)$ is a decreasing function around zero. For instance, when $h_1(x) = h_3(x) = \log(\sigma(x))$ and $h_2(x) = \log(1 - \sigma(x))$ with $\sigma(\cdot)$ denoting the sigmoid function, we obtain the vanilla GAN (Goodfellow et al., 2014).

Since θ and ϕ are updated using gradient descent, we can denote the training trajectories as signals $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. The dynamics are defined by the following gradient flow:

$$\begin{aligned} \frac{d\boldsymbol{\phi}(t)}{dt} &= \frac{\partial \mathcal{V}_1(\phi; \theta)}{\partial \phi}|_{\phi=\boldsymbol{\phi}(t), \theta=\boldsymbol{\theta}(t)}, \\ \frac{d\boldsymbol{\theta}(t)}{dt} &= \frac{\partial \mathcal{V}_2(\theta; \phi)}{\partial \theta}|_{\phi=\boldsymbol{\phi}(t), \theta=\boldsymbol{\theta}(t)}. \end{aligned} \tag{7}$$

Specifically, for the dynamics of $\mathcal{D}$, we have:

$$\frac{\partial \mathcal{V}_1(\phi; \theta)}{\partial \phi} = \frac{dh_1(\mathcal{D}(c))}{d\phi} + \frac{dh_2(\mathcal{D}(\theta))}{d\phi}. \tag{8}$$

Similarly, for the dynamics of $\mathcal{G}$, we have:

$$\frac{\partial \mathcal{V}_2(\theta; \phi)}{\partial \theta} = \frac{dh_3(\mathcal{D}(\theta))}{d\mathcal{D}(\theta)} \frac{\partial \mathcal{D}(\theta)}{\partial \theta}. \tag{9}$$

Substituting $\mathcal{D}(x) = \phi x$ to Eqn. (8) and Eqn. (9), the dynamics of Dirac GAN can be summarized as:

$$\begin{aligned} \frac{d\boldsymbol{\phi}(t)}{dt} &= h_1'(\boldsymbol{\phi}(t)c)c + h_2'(\boldsymbol{\phi}(t)\boldsymbol{\theta}(t))\boldsymbol{\theta}(t), \\ \frac{d\boldsymbol{\theta}(t)}{dt} &= h_3'(\boldsymbol{\phi}(t)\boldsymbol{\theta}(t))\boldsymbol{\phi}(t), \end{aligned} \tag{10}$$

where $h_i'(\cdot)$ denotes the derivative of $h_i(\cdot)$ for $i \in \{1, 2, 3\}$.

From the perspective of control theory (see details in Sec. 2.1), Eqn. (10) represents a dynamic in the time domain, which is natural to understand but difficult to analyze. Converting it to the frequency domain by the Laplace transform (LT) can simplify the analysis. It requires a case by case derivation for different GANs due to the specific forms of the objective functions (i.e., different choices of $h_i(\cdot)$). We will first use WGAN as an example to present the analyzing process and then generalize it to other objectives via the local linearization technique in Sec. 3.3.

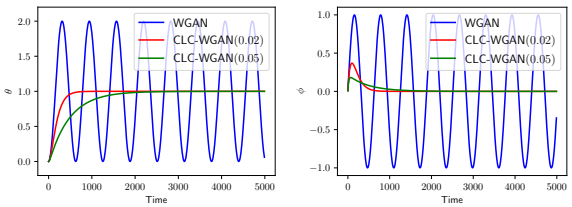

Figure 1: The simulated dynamic of Dirac GAN for $\boldsymbol{\theta}$ (left) and $\boldsymbol{\phi}$ (right) with $c = 1$. The curve of WGAN shows the oscillation while Other curves of CLC-GAN show that the closed loop control helps convergence.

In WGAN[4], we have $h_1(x) = h_3(x) = x$ and $h_2(x) = -x$. Let the output $\boldsymbol{y}(t) = (\boldsymbol{\theta}(t), \boldsymbol{\phi}(t))$ and the input $\boldsymbol{u}(t) = c,\ \forall t > 0$. Then, the dynamic in Eqn. (8) and Eqn. (9) is instantiated as:

$$\frac{d\boldsymbol{y}(t)}{dt} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}(t) \\ \boldsymbol{\phi}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{u}(t) \end{bmatrix} = f(\boldsymbol{y}(t), \boldsymbol{u}(t)). \tag{11}$$

Applying LT $\mathcal{F}(\cdot)$ in Eqn. (2) to both sides of Eqn. (11), the dynamic can be represented in the frequency domain as:

$$\begin{cases} s\boldsymbol{\Phi}(s) = \boldsymbol{U}(s) - \boldsymbol{\Theta}(s), \\ s\boldsymbol{\Theta}(s) = \boldsymbol{\Phi}(s). \end{cases} \tag{12}$$

where $\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{U}$ represent $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{u}$ in the frequency domain, e.g., $\boldsymbol{U}(s) = \mathcal{F}(\boldsymbol{u})(s)$. Then we can solve the dynamics of $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ according to the formal rules of algebra as:

$$\begin{cases} \boldsymbol{\Phi}(s) = \frac{s}{s^2+1}\boldsymbol{U}(s), \\ \boldsymbol{\Theta}(s) = \frac{1}{s}\boldsymbol{\Phi}(s) = \frac{1}{s^2+1}\boldsymbol{U}(s). \end{cases} \tag{13}$$

In the frequency domain, the output signal can be represented as a multiplication between the transfer function (see Sec. 2.1) and the input signal. Specifically, in Eqn. (13), the transfer function of $\boldsymbol{\phi}$ is $\boldsymbol{T}_{\mathcal{D}}(s) = \frac{s}{s^2+1}$ and the transfer function of $\boldsymbol{\theta}$ is $\boldsymbol{T}_{\mathcal{G}}(s) = \frac{1}{s^2+1}$. According to Proposition 1, the stability of a dynamic is fully characterized by the poles of the transfer function (i.e., the roots of the denominator). The poles of both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are $\pm i$ according to Eqn. (13). Therefore, both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are oscillatory instead of converging to the equilibrium point $(\boldsymbol{\theta}_e, \boldsymbol{\phi}_e) = (c, 0)$. The simulated dynamic of Dirac GAN is illustrated in Fig. 1.

3.2. Analyzing and Improving Stability

Control theory provides extensive methods (Khalil, 2002) to improve the stability of dynamics without changing the desired equilibrium. In this paper, the widely used closed-loop control (CLC) is introduced in Sec. 2.3 for its simplicity. We emphasize that advanced control methods can potentially result in more stable GANs and we leave it as future work.

Before applying the CLC, we emphasize that there are two requirements to be satisfied simultaneously: 1) applying the CLC needs to stabilize the dynamics of $\mathcal{D}$ and $\mathcal{G}$; 2) it should not change the equilibrium point of $\mathcal{G}$, i.e., $p_{\mathcal{G}} = p$.

For the first requirement, the dynamic of $\boldsymbol{\theta}$ in Dirac GAN is $\frac{d\boldsymbol{\theta}(t)}{dt} = h_3'(\boldsymbol{\phi}(t)\boldsymbol{\theta}(t))\boldsymbol{\phi}(t)$, which indicates that stabilizing $\boldsymbol{\phi}$ to zero can also stabilize the dynamic of $\boldsymbol{\theta}$. Therefore, we only need to introduce the CLC to $\mathcal{D}$. The central idea of the CLC is to adjust the transfer function by introducing an auxiliary controller. Here we adopt a simple and widely used controller $\boldsymbol{T}_b(s) = \lambda$. Intuitively, it is an amplifier with negative feedback from output to input according to Eqn. (4) and λ[5] is the coefficient for the amplitude of the feedback. Substituting $\boldsymbol{T}_b$ with λ in Eqn. (5), the transfer function $\boldsymbol{T}_{c\mathcal{D}}$ of the controlled $\boldsymbol{\phi}$ is given by:

$$\boldsymbol{T}_{c\mathcal{D}}(s) = \frac{\frac{s}{s^2+1}}{1 + \frac{\lambda s}{s^2+1}} = \frac{s}{s^2 + \lambda s + 1}. \tag{14}$$

With a positive λ, all of poles in the controlled dynamic have negative real parts, and hence it is a stable dynamic. We also demonstrate the simulated results of the controlled dynamic with different values of λ in Fig. 1.

For the second requirement, the CLC will not change the equilibrium point of Dirac GAN. In the time domain, the CLC is equivalent to adjust the dynamics of $\boldsymbol{\phi}$ as:

$$\frac{d\boldsymbol{\phi}}{dt} = c - \boldsymbol{\theta}(t) - \lambda\boldsymbol{\phi}(t). \tag{15}$$

Since the equilibrium point of $\mathcal{D}$ is a zero function, i.e., $\boldsymbol{\phi}_e = 0$, then we still have $\frac{d\boldsymbol{y}(t)}{dt} = 0$ at $\boldsymbol{y} = (\boldsymbol{\theta}_e, \boldsymbol{\phi}_e)$.

3.3. Extending to Other Objectives

The proposed method is not limited to WGAN but can be generalized to other GANs (Goodfellow et al., 2014; Mao et al., 2017), which may have nonlinear objective functions.

We leverage a standard technique called *local linearization* (Khalil, 2002) to approximate the original dynamics as a linear one around the equilibrium point. For example, the objective function of $\mathcal{D}$ in the vanilla GAN is:

$$\max_{\boldsymbol{\phi}} \mathcal{V}_s(\phi, \theta) = \log(\sigma(\phi c)) + \log(1 - \sigma(\phi\theta)). \tag{16}$$

The dynamic of $\boldsymbol{\phi}$ is nonlinear because of the sigmoid func-

[4]We ignore the Lipschitz continuity of $\mathcal{D}$ for simplicity but the equilibrium point and its local convergence do not change. See theoretical analysis and empirical evidence in Appendix B.

[5]λ is a hyperparameter and we analyze its sensitivity in Sec. 6.

Table 1: The stability characters for the widely-used GANs. Please refer to Appendix A for detailed derivation, which adopts the local linearization technique introduced in Sec. 3.3. With CLC, the training dynamics of Dirac GANs are stable theoretically (see Fig. 1 and Appendix A), and those of normal GANs are stable empirically (see Fig. 2).

	$\boldsymbol{T}_{\mathcal{D}}(s)$	Stability Dirac GAN/normal GAN	$\boldsymbol{T}_{c\mathcal{D}}(s)$	Stability with CLC Dirac GAN/normal GAN
WGAN	$s/(s^2+1)$	✗/✗	$1/(s^2+\lambda s+1)$	✓/✓
Hinge-GAN	$s/(s^2+1)$	✗/✗	$1/(s^2+\lambda s+1)$	✓/✓
SGAN	$2s/(4s^2+2s+1)$	✓/✗	$1/(4s^2+(2\lambda+2)s+1)$	✓/✓
LSGAN	$s/(s^2+4s+1)$	✓/✗	$1/(s^2+(\lambda+4)s+1)$	✓/✓

tion, which is given by:

$$\frac{d\boldsymbol{\phi}(t)}{dt} = \frac{\partial \mathcal{V}_s(\phi,\theta)}{\partial \phi}|_{\phi=\boldsymbol{\phi}(t),\theta=\boldsymbol{\theta}(t)} = \frac{\sigma'(\boldsymbol{\phi}(t)c)}{\sigma(\boldsymbol{\phi}(t)c)}c - \frac{\sigma'(\boldsymbol{\phi}(t)\boldsymbol{\theta}(t))}{1-\sigma(\boldsymbol{\phi}(t)\boldsymbol{\theta}(t))}\boldsymbol{\theta}(t), \tag{17}$$

where $\sigma'(\cdot)$ is the derivative of $\sigma(\cdot)$. Local linearization approximates the original dynamic by the first order Taylor expansion at the equilibrium point $(c, 0)$:

$$\frac{\partial \mathcal{V}_s(\phi,\theta)}{\partial \phi} \approx \frac{\partial \mathcal{V}_s(\phi,\theta)}{\partial \phi}|_{\phi=0,\theta=c} + \frac{\partial^2 \mathcal{V}_s(\phi,\theta)}{\partial \phi^2}|_{\phi=0,\theta=c}\phi + \frac{\partial^2 \mathcal{V}_s(\phi,\theta)}{\partial \theta \partial \phi}|_{\phi=0,\theta=c}(\theta-c) = -\frac{1}{2}\phi - \frac{1}{2}(\theta-c). \tag{18}$$

Note that the stability is determined by the local character of the equilibrium point, around which the residual in Eqn. (18) is negligible. Therefore, we have a linear approximation and the the analysis in Sec. 3.2 applies. We summarize the stability characters for all GANs in Table 1.

4. Extensions to Normal GANs

In Sec. 3, we show that the dynamic of Dirac GAN can be formally analyzed and stabilized based on the recipe for control theory. Besides, the CLC can successfully stabilize nonlinear dynamics in control theory (Khalil, 2002). This two facts inspire us to stabilize the training dynamic of a normal GAN (i.e., parameterized by neural networks) by incorporating the CLC. Unlike previous methods (Mescheder et al., 2018) which mainly focus on the dynamics of parameters of $\mathcal{D}$ and $\mathcal{G}$, we instead model the dynamics of $\mathcal{G}$ and $\mathcal{D}$ in the function space, i.e., $\mathcal{D} = \mathcal{D}(x,t)$ and $\mathcal{G} = \mathcal{G}(z,t)$. It can simplify the analysis and build the connections between the Dirac GAN and the normal GANs.

Following the notation in Sec. 3, the objective function of a general GAN is:

$$\max_{\mathcal{D}} \mathcal{V}_1(\mathcal{D};\mathcal{G}) = \mathbb{E}_{p(x)}[h_1(\mathcal{D}(x))] + \mathbb{E}_{p_{\mathcal{G}}(x)}[h_2(\mathcal{D}(x))],$$
$$\max_{\mathcal{G}} \mathcal{V}_2(\mathcal{G};\mathcal{D}) = \mathbb{E}_{p_z(z)}[h_3(\mathcal{D}(\mathcal{G}(z)))]. \tag{19}$$

According to the calculus of variations (Gelfand et al., 2000), the gradient of $\mathcal{V}_1(\mathcal{D})$ with respect to the function $\mathcal{D}$ is:

$$\frac{\partial \mathcal{V}_1(\mathcal{D};\mathcal{G})}{\partial \mathcal{D}} = p\frac{dh_1(\mathcal{D})}{d\mathcal{D}} + p_{\mathcal{G}}\frac{dh_2(\mathcal{D})}{d\mathcal{D}}, \tag{20}$$

where $\frac{dh_i(\mathcal{D})}{d\mathcal{D}}(x) = \frac{dh_i(u)}{du}|_{u=\mathcal{D}(x)} = \frac{dh_i(\mathcal{D}(x))}{d\mathcal{D}(x)}$ for $i \in \{1,2\}$. The gradient of $\mathcal{V}_2(\mathcal{G})$ with respect to $\mathcal{G}$ is:

$$\frac{\partial \mathcal{V}_2(\mathcal{G})}{\partial \mathcal{G}} = p_z\frac{dh_3(\mathcal{D}(\mathcal{G}))}{d\mathcal{G}}, \tag{21}$$

where $\frac{dh_3(\mathcal{D}(\mathcal{G}))}{d\mathcal{G}}(z) = \frac{dh_3(\mathcal{D}(\mathcal{G}(z)))}{d\mathcal{D}(\mathcal{G}(z))}\frac{\partial \mathcal{D}(\mathcal{G}(z))}{\partial \mathcal{G}(z)}$.

Therefore, the dynamics of $\mathcal{D}$ and $\mathcal{G}$ in normal GANs can

Algorithm 1 Closed-loop Control GAN

Input: Buffer size N_b, feedback coefficient λ, batch size N, initialized $\mathcal{G}$ and $\mathcal{D}$, learning rate η.
Initialize B_r and B_f for real samples and fake samples respectively.
repeat
  Sample a batch of $\{x_r\} \sim p$, $\{x_f\} \sim p_{\mathcal{G}}$ of N samples.
  Update B_r with $\{x_r\}$. Update B_f with $\{x_f\}$.
  Sample a batch of $x'_r \sim B_r$, $x'_f \sim B_f$ of N samples respectively.
  Estimate the objective of $\mathcal{D}$: $\mathcal{U}(\mathcal{D}) = \frac{1}{N}[\sum_{x\in\{x_r\}}\mathcal{D}(x) - \sum_{x\in\{x_f\}}\mathcal{D}(x)] - \frac{\lambda}{N}[\sum_{x\in\{x'_r\}}\mathcal{D}^2(x) + \sum_{x\in\{x'_f\}}\mathcal{D}^2(x)]$.
  Update $\mathcal{D}$ to maximize $\mathcal{U}(\mathcal{D})$ with learning rate η.
  Estimate the objective of $\mathcal{G}$: $\mathcal{U}(\mathcal{G}) = \frac{1}{N}\sum_{x\in\{x_f\}}\mathcal{D}(x)$.
  Update $\mathcal{G}$ to maximize $\mathcal{U}(\mathcal{G})$ with learning rate η.
until Convergence

be denoted generally as:

$$\frac{d\mathcal{D}(x,t)}{dt} = p(x)\frac{dh_1(\mathcal{D}(x))}{d\mathcal{D}(x,t)} + p_{\mathcal{G}}(x)\frac{dh_2(\mathcal{D}(x))}{d\mathcal{D}(x)}, \forall x,$$
$$\frac{d\mathcal{G}(z,t)}{dt} = p_z(z)\frac{dh_3(\mathcal{D}(\mathcal{G}(z)))}{d\mathcal{D}(\mathcal{G}(z))}\frac{\partial\mathcal{D}(\mathcal{G}(z))}{\partial\mathcal{G}(z)}, \forall z. \quad (22)$$

Note that the above dynamics is quiet similar to the dynamic of Dirac GAN by substituting $\mathcal{G}$ and $\mathcal{D}$ for θ and ϕ in Eqn. (8) and Eqn. (9) respectively. Specifically, in both dynamics, the discriminators take the weighted summation of p and $p_{\mathcal{G}}$. For the generator, both of them depend on the $\frac{\partial\mathcal{D}(\mathcal{G}(z))}{\partial\mathcal{G}(z)}$. The above similarity between Dirac GANs and normal GANs inspires us to directly apply the CLC in nonlinear settings. Our empirical results in various settings (see Sec. 6) demonstrate the effectiveness of the proposed method, which agrees with the above analysis and Table 1.

4.1. Implementing CLC in GANs

According to Sec. 3.2, we apply the CLC with a controller $\boldsymbol{T}_b(s) = \lambda$ to normal GANs. The resulting dynamic of $\mathcal{D}$ is

$$\frac{d\mathcal{D}(x,t)}{dt} = \frac{\partial\mathcal{V}_1(\mathcal{D};\mathcal{G})}{\partial\mathcal{D}} - \lambda\mathcal{D}(x), \forall x. \quad (23)$$

Note that $\mathcal{D}$ will be optimized by gradient descent in the implementation and we need to design a proper objective function whose gradient flow is equivalent to Eqn. (23). Therefore, we introduce an auxiliary regularization term to the original GANs and get:

$$\mathcal{V}_1'(\mathcal{D};\mathcal{G}) = \mathcal{V}_1(\mathcal{D};\mathcal{G}) - \frac{\lambda}{2}\int_{x\in\mathcal{X}}\mathcal{D}^2(x)dx, \quad (24)$$

where $\mathcal{X}$ denotes the space of x, e.g., $\mathcal{X} = [-1,1]^{c\times w\times h}$ for image generation of size $w \times h \times c$. Below, we denote $\mathcal{R}(\mathcal{D}) = \int_{x\in\mathcal{X}}\mathcal{D}^2(x)dx$, which is the squared 2-norm of the function $\mathcal{D}$ over the space of x. Intuitively, minimizing $\mathcal{R}(\mathcal{D})$ encourages $\mathcal{D}$ to converge to a zero function.

The regularization term $\mathcal{R}(\mathcal{D})$ is proportional to the expectation of $\mathcal{D}^2$ with respect to a uniform distribution $p_u(x)$ defined on $\mathcal{X}$, i.e., $\mathcal{R}(\mathcal{D}) \propto \mathbb{E}_{p_u(x)}[\mathcal{D}^2(x)]$. However, directly estimating $\mathcal{R}(\mathcal{D})$ is not sample efficient since most of samples in $\mathcal{X}$ is meaningless and do not provide useful training signals to stabilize $\mathcal{D}$. Instead, we maintain two buffers B_r and B_f of fix size N_b to store the old real samples and fake samples, respectively. We define a uniform distribution $p_u^t(x)$ on $B^t = B_r^t \cup B_f^t$ to approximate $\mathcal{R}(\mathcal{D})$ as:

$$\mathcal{R}_t(\mathcal{D}) = \int_{x\in\mathcal{X}} p_u^t(x)\mathcal{D}^2(x)dx. \quad (25)$$

where $\mathcal{R}_t(\mathcal{D})$ denotes the regularization term at time t. $\mathcal{R}_t(\mathcal{D})$ is estimated using Monte Carlo and these buffers are updated with replacement. As analyzed below, using $\mathcal{R}_t(\mathcal{D})$ to approximate $\mathcal{R}(\mathcal{D})$ will not change the equilibrium and stability. The training procedure is presented in Alg. 1.

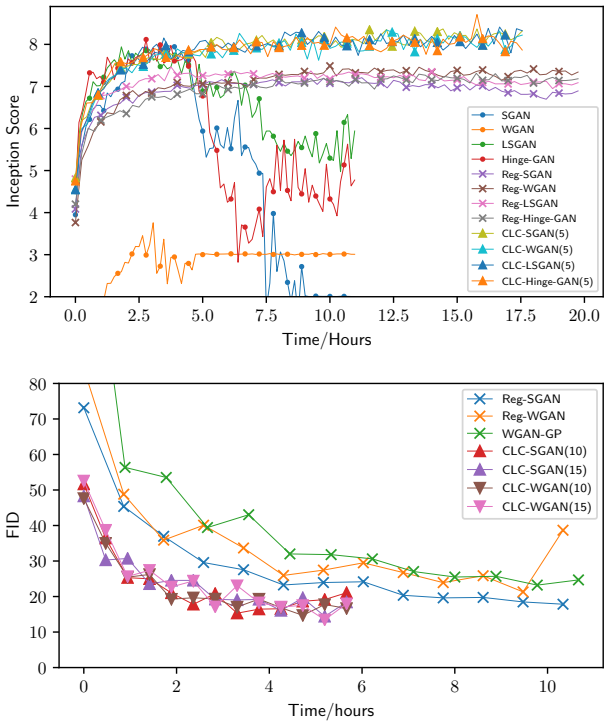

Figure 2: The learning curve of the baselines and our proposed method. Top: The Inception Score of CIFAR10. Bottom: The FID score of CelebA. We plot the curves with respect to the time for better representation of the computational cost.

4.2. Theoretical Analysis

Below, we first prove that the regularization term in Eqn. (25) will not change the desirable equilibrium point of GANs, i.e., $p_{\mathcal{G}} = p$, as summarized in Lemma 1.

Lemma 1. *Under the non-parametric setting, CLC-GAN has the same equilibrium as the original GAN, i.e, $p_{\mathcal{G}} = p$ and $\mathcal{D}(x) = 0$ for all x.*

Here we follow the identical assumption as in Goodfellow et al. (2014). Further, under mild assumptions as in Mescheder et al. (2018), CLC-GAN locally converges to the equilibrium, as summarized in Theorem 1.

Theorem 1. *(Proof in Appendix C) Under the Assumptions 1, 2 and 3 in Appendix C with sufficient small learning rate and large λ, the parameters of CLC-GAN locally converge to the equilibrium with alternative gradient descent.*

We provide the experimental results in Sec. 6 to empirical validate our method.

5. Related Work

Some recent work directly models the training process of GANs. Mescheder et al. (2017) and Nagarajan & Kolter (2017) model the dynamics of GANs in the parameter space and stabilize the training dynamics using gradient-based regularization. However, the above methods do not model the whole training dynamics explicitly and cannot generalize to natural images. Then Mescheder et al. (2018) propose a prototypical example Dirac GAN to understand GANs' training dynamics and stabilize GANs using simplified gradient penalties. Gidel et al. (2018) analyze the effect of momentum based on the Dirac GAN and propose the negative momentum. Though the above methods provide an elegant understanding of the training dynamics, this understanding does not provide a practically effective algorithm to stabilize nonlinear GANs' training and they fail to report competitive results to the state-of-the-art (SOTA) methods (Miyato et al., 2018). Instead, we revisits the Dirac GAN from the perspective of control theory, which provides a set of tools and extensive experience to stabilize it. Based on the recipe, we advance the previous SOTA results on image generation.

Feizi et al. (2017) is another related work that analyzes the stability of GANs using the Lyapunov function, which is a general approach in control theory. However, it only focuses on the stability analysis whereas cannot provide stabilizing methods. In our paper, we are interested in building SOTA GANs in practise and therefore we leverage the classical control theory.

6. Experiments

We now empirically verify our method on the widely-adopted CIFAR10 (Krizhevsky et al., 2009) and CelebA (Liu et al., 2015) datasets. CIFAR10 consists of 50,000 natural images of size 32×32 and CelebA consists of 202,599 face images of size 64×64. The quantitative results are from the corresponding papers or reproduced on the official code for fair comparison. Specifically, we use the exactly same architectures for both $\mathcal{D}$ and $\mathcal{G}$ with our baseline methods, where the ResNet (He et al., 2016) with the ReLU activation (Glorot et al., 2011) is adopted[6]. The batch size is 64, and the buffer size N_b is set to be 100 times of the batch size for all settings. We manually select the coefficient λ among $\{1, 2, 5, 10, 15, 20\}$ in Reg-GAN's setting and among $\{0.05, 0.1, 0.2, 0.5\}$ in SN-GAN's setting. We use the Inception Score (IS) (Salimans et al., 2016) to evaluate the image quality on CIFAR10 and FID score (Gulrajani et al., 2017) on both CIFAR10 and CelebA. More details about the experimental setting and further results on a synthetic dataset can be found in Appendix E.

[6]Our code is provided HERE.

Table 2: The FID Score on CIFAR10. The results reported here are the best results over the training process and are averaged over 3 runs.

Method	WGAN	SGAN
No Regularization	105.21	28.51
Reg-GAN	30.43	28.39
Gradient Penalty	28.20	–
CLC-GAN(2)	23.53 ± 1.22	21.63 ± 0.47
CLC-GAN(5)	21.46 ± 1.57	$\mathbf{21.52} \pm 0.96$
CLC-GAN(10)	$\mathbf{21.14} \pm 1.84$	22.20 ± 2.07

Table 3: The Inception score on CIFAR10. † (Yang et al., 2017), ‡ (Miyato et al., 2018), § (Zhang et al., 2019). Results of CLC-GAN are averages over 3 runs.

Method	WGAN	SGAN	Hinge
LR-GAN†	-	7.17	-
SN-GAN‡	-	-	8.22
CR-GAN§	-	8.40	-
Gradient Penalty	7.82	-	-
Reg-GAN	7.34	7.37	7.37
CLC-GAN(2)	$8.42 \pm .06$	$8.28 \pm .05$	$8.49 \pm .08$
CLC-GAN(5)	$\mathbf{8.49} \pm .07$	$8.44 \pm .08$	$\mathbf{8.54} \pm .03$
CLC-GAN(10)	$8.38 \pm .10$	$\mathbf{8.47} \pm .09$	$8.46 \pm .00$
SN-GAN	3.29	8.17	8.28
CLC-SN-GAN(0.1)	$8.14 \pm .02$	$8.30 \pm .09$	$\mathbf{8.54} \pm .03$

We compare with two typical families of GANs. The first one is referred as unregularized GANs, including WGAN (Arjovsky et al., 2017), SGAN (Goodfellow et al., 2014), LSGAN (Mao et al., 2017) and Hinge-GAN (Miyato et al., 2018).The second one is referred as reguarlized GANs, including Reg-GAN (Mescheder et al., 2018) and SN-GAN (Miyato et al., 2018). We emphasize that the regularzied GANs are the previous SOTA methods and our implementations are based on the officially released code. For clarity, we refer to our method as CLC-GAN$(\cdot)$ with the hyperparameter λ denoted in the parentheses.

In the following, we will demonstrate that (1) the CLC can stabilize GANs using less computational cost than competitive regularizations and is applicable to various objective functions ; (2) CLC-GAN provides a consistent improvement on the quantitative results in different settings compared to related work (Mescheder et al., 2018) and surpasses previous state-of-the-art (SOTA) GANs (Miyato et al., 2018; Zhang et al., 2019).

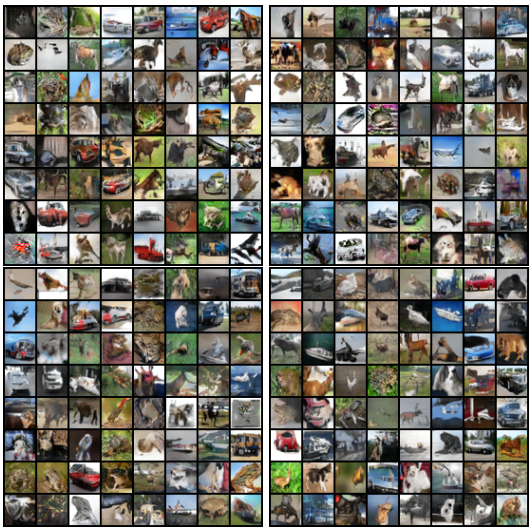

Figure 3: The generated results of CIFAR10 dataset. From top left to bottom right: WGAN-GP, Reg-WGAN, CLC-WGAN(5), CLC-SGAN(5).

Figure 4: The generated results of CelebA dataset. From top left to bottom right: WGAN-GP, Reg-WGAN, CLC-WGAN(15), CLC-SGAN(15).

6.1. CLC-GAN is stable

In the linear case, the simulated results in Fig. 1 demonstrate that CLC-GAN can stabilize the Dirac GAN, which agrees with our theoretical analysis in Sec. 3.2.

In normal GANs, we compare CLC-GAN with a wide range of GANs (Arjovsky et al., 2017; Goodfellow et al., 2014; Mao et al., 2017; Miyato et al., 2018) and their regularized version in (Mescheder et al., 2018) in terms of training stability qualitatively. The learning curves are shown in Fig. 2. The top panel shows the IS on CIFAR10 and the bottom one shows FID on CelebA.

In both panels, the training dynamics of unregularized GANs are not stable. On CIFAR10, the unregularized GANs all diverge from the data distribution and on CelebA they even diverge at the very beginning. Indeed, their FID results on CelebA are over 300 which is too large to be shown in the figure. Among unregularized GANs, LSGAN and SGAN are more stable than WGAN on CIFAR10 which is consistent to our analysis in Table 1. However, none of them provide converged results, nor can they generalize to larger images in CelebA. We hypothesize that the nonlinearity in neural networks is the main reason for the divergence behaviour. Instead, CLC-GAN can succetssfully avoid the oscillatory behaviour and regularize GANs towards the data distribution. The robustness of CLC-GAN in the nonlinear dynamics agrees with the theoretical analysis in Table 1 and the experience in control theory, which are the main motivations of our paper. In conclusion, the comparison between the unregularized GANs and their controlled versions show the effectiveness of the proposed method.

Indeed, Reg-GAN can also stabilize the training dynamics. In comparison, the CLC-GANs are computationally efficient and achieve better results after convergence. First, unlike the gradient penalty which implies a non-trivial running time (Kurach et al., 2018), CLC-GANs directly regularize the activation of $\mathcal{D}$ and require less computational cost. For instance, our method can conduct approximate 8 iterations per second of training on CelebA whereas Reg-GAN can only conduct 4 iterations per second on Geforce 1080Ti. Second, CLC-GANs provide higher IS on CIFAR10 and lower FID on CelebA as qualitatively shown in the learning curves. The quantitative results are summarized in the following subsection.

Fig. 3 & Fig. 4 show the generated samples. Those from CLC-GAN are semantically meaningful in all setting and are at least competitive to the ones from very strong baselines.

6.2. Quantitative Results

We now present the quantitative results on CIFAR10 in the settings that include different objective functions, neural network architectures and the values of λ. The IS and FID are shown in Table 3 and Table 2 respectively. The comparisons among different settings are given within the tables.

First, our method provides a consistent improvements on both IS and FID on CIFAR10. For FID, CLC-GANs decrease it from 28 to 23 compared to Reg-GAN. For IS, CLC-GANs surpass previous SOTA GANs. Specifically, CLC-GANs achieve IS over 8.45 with various objectives without using spectral normalization, which is a significant improvement compare to related works, including SN-GAN (Miyato

et al., 2018) and CR-GAN (Zhang et al., 2019).

Second, CLC-GAN is also applicable to SN-GAN's architecture and improve its performance, whereas most gradient-based regularizations fail to introduce significant improvement (Kurach et al., 2018). Unlike SN-GAN whose performance largely depends on the objective functions, CLC-SN-GAN provides stable training dynamics consistently.

Finally, CLC-GAN is not very sensitive to the hyperparameter λ given the normalization used in $\mathcal{D}$. When batch normalization is adopted, CLC-GANs with $\lambda = 2, 5, 10$ all achieve SOTA IS and a large improvement on FID. When spectral normalization (Miyato et al., 2018) is used, a relatively smaller λ is required. Besides the reported results with $\lambda = 0.1$, CLC-SN-GANs with $\lambda \in \{0.05, 0.2\}$ achieves IS over 8.4 consistently using Hinge loss. The underlying mechanism of the difference between the two types of normalizations is unclear. We hypothesize that it is because $\mathcal{D}$ is a Lipschitz-1 function with spectral normalization.

7. Conclusions and Discussions

In this paper, we propose a novel perspective to understand the dynamics of GANs and a stabilizing method called CLC-GAN. We model the dynamics of the Dirac GAN with linear objectives theoretically in the frequency domain and extend the analysis to nonlinear objectives using local linearization. By leveraging the recipe for control theory, we propose a stabilizing method called CLC to improve Dirac GAN's stability and generalize CLC to normal GANs. The simulated results on Dirac GAN and empirical results on normal GANs demonstrate that our method can stabilize a wide range of GANs and provide better convergence results.

Although CLC-GAN provides promising results, further analyses can be done to achieve better results. On one hand, our analysis mainly focuses on the continuous cases, where the practical implementation optimizes both $\mathcal{G}$ and $\mathcal{D}$ in discrete time steps. In this case, the $\mathcal{Z}$-transform is a better tool than LT used in this paper. On the other hand, we approximate the dynamics in the function space using the update in the parameter space, which can be improved by recent analyses of GANs in the function space (Johnson & Zhang, 2018). Finally, modern control theory and nonlinear control methods (Khalil, 2002) can potentially help GANs to achieve better performance. These are promising directions for the future work.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2017YFA0700904), NSFC Projects (Nos. 61620106010, U19B2034, U181146), Beijing NSF Project (No. L172037), Beijing Academy of Artificial Intelligence, Tsinghua-Huawei Joint Research Program, Tiangong Institute for Intelligent Computing, and NVIDIA NVAIL Program with GPU/DGX Acceleration. C. Li was supported by the Chinese postdoctoral innovative talent support program and Shuimu Tsinghua Scholar.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223, 2017.

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

Chavdarova, T. and Fleuret, F. Sgan: An alternative training of generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9407–9415, 2018.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Donahue, J. and Simonyan, K. Large scale adversarial representation learning. *arXiv preprint arXiv:1907.02544*, 2019.

Du, C., Xu, K., Li, C., Zhu, J., and Zhang, B. Learning implicit generative models by teaching explicit ones. *arXiv preprint arXiv:1807.03870*, 2018.

Feizi, S., Farnia, F., Ginart, T., and Tse, D. Understanding gans: the lqg setting. *arXiv preprint arXiv:1710.10793*, 2017.

Gelfand, I. M., Silverman, R. A., et al. *Calculus of variations*. Courier Corporation, 2000.

Gidel, G., Hemmat, R. A., Pezeshki, M., Lepriol, R., Huang, G., Lacoste-Julien, S., and Mitliagkas, I. Negative momentum for improved game dynamics. *arXiv preprint arXiv:1807.04740*, 2018.

Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, 2011.

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

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

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

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

Johnson, R. and Zhang, T. Composite functional gradient learning of generative adversarial models. *arXiv preprint arXiv:1801.06309*, 2018.

Kailath, T. *Linear systems*, volume 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

Khalil, H. K. Nonlinear systems. *Upper Saddle River*, 2002.

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

Kurach, K., Lucic, M., Zhai, X., Michalski, M., and Gelly, S. A large-scale study on regularization and normalization in gans. *arXiv preprint arXiv:1807.04720*, 2018.

Li, C., Xu, T., Zhu, J., and Zhang, B. Triple generative adversarial nets. In *Advances in neural information processing systems*, pp. 4088–4098, 2017.

Liang, K. J., Li, C., Wang, G., and Carin, L. Generative adversarial network training is a continual learning problem. *arXiv preprint arXiv:1811.11083*, 2018.

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

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, 2017.

Mescheder, L., Nowozin, S., and Geiger, A. The numerics of gans. In *Advances in Neural Information Processing Systems*, pp. 1825–1835, 2017.

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

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

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

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Widder, D. V. *Laplace transform (PMS-6)*. Princeton university press, 2015.

Yang, J., Kannan, A., Batra, D., and Parikh, D. Lr-gan: Layered recursive generative adversarial networks for image generation. *arXiv preprint arXiv:1703.01560*, 2017.

Zhang, H., Zhang, Z., Odena, A., and Lee, H. Consistency regularization for generative adversarial networks. *arXiv preprint arXiv:1910.12027*, 2019.