Appendix: Understanding and Stabilizing GANs' Training Dynamics Using Control Theory

Anonymous Authors[1]

A. Dynamics for different GANs.

In this section, we apply the local linearization technique to Dirac GANs with various objective functions, including vanilla GAN, non-saturation GAN (Goodfellow et al., 2014), LS-GAN (Mao et al., 2017) and Hinge-GAN (Miyato et al., 2018). Following the notations in the main body, the training dynamics of general Dirac GANs are given by:

$$\frac{d\phi(t)}{dt} = \frac{\partial V_1(\phi;\theta)}{\partial \phi} = h_1'(\phi c)c - h_2'(\phi\theta)\theta, \tag{A.1}$$

$$\frac{d\theta(t)}{dt} = \frac{\partial V_2(\theta;\phi)}{\partial \theta} = h_3'(\phi\theta)\phi. \tag{A.2}$$

By applying the local linearization technique to both ϕ and θ around the equilibrium point $(\phi_c, \theta_c) = (0, c)$, the dynamic can be approximated as:

$$\begin{bmatrix} \frac{d\phi(t)}{dt} \\ \frac{d\theta(t)}{dt} \end{bmatrix} \approx \begin{bmatrix} \frac{\partial^2 V_1(\phi;\theta)}{\partial \phi^2} & \frac{\partial^2 V_1(\phi;\theta)}{\partial \theta \partial \phi} \\ \frac{\partial^2 V_2(\phi;\theta)}{\partial \phi \partial \theta} & \frac{\partial^2 V_2(\phi;\theta)}{\partial \theta^2} \end{bmatrix} \begin{bmatrix} \phi(t) - \phi_e \\ \theta(t) - \theta_e \end{bmatrix} \tag{A.3}$$

$$= \mathbf{T} \begin{bmatrix} \phi(t) - \phi_e \\ \theta(t) - \theta_e \end{bmatrix} = \mathbf{T} \begin{bmatrix} \phi(t) \\ \theta(t) - c(t) \end{bmatrix}, \tag{A.4}$$

and $\mathbf{T}$ can be denoted as:

$$\mathbf{T} = \begin{bmatrix} h_1''(\phi c)c^2 + h_2''(\phi\theta)\theta^2 & h_2'(\phi\theta) + h_2''(\phi\theta)\theta\phi \\ h_3'(\phi\theta) + h_3''(\phi\theta)\phi\theta & h_3''(\phi\theta)\phi^2 \end{bmatrix}.$$

Here $h_i''(x)$ is the second order derivative of $h_i(x)$ for $i \in \{1, 2, 3\}$. Below we assume $c = 1$ and start the case by case analysis for various types of GANs.

A.1. Vanilla GAN

In vanilla GAN, we have:

$$h_1(x) = \log(\sigma(x)), \tag{A.5}$$

$$h_2(x) = \log(1 - \sigma(x)), \tag{A.6}$$

$$h_3(x) = -\log(1 - \sigma(x)), \tag{A.7}$$

where $\sigma(\cdot)$ denotes the sigmoid function. Then we have:

$$h_1'(x) = (1 - \sigma(x)), h_1''(x) = -\sigma(x)(1 - \sigma(x)), \tag{A.8}$$

$$h_2'(x) = -\sigma(x), h_2''(x) = -\sigma(x)(1 - \sigma(x)), \tag{A.9}$$

$$h_3'(x) = \sigma(x), h_3''(x) = \sigma(x)(1 - \sigma(x)). \tag{A.10}$$

and for $\mathbf{T}$:

$$\mathbf{T} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix}. \tag{A.11}$$

It indicates that

$$\Phi(s) = -\frac{1}{2s+1}(\Theta(s) - C(s)) \tag{A.12}$$

$$\Theta(s) = \frac{1}{2s}\Phi(s). \tag{A.13}$$

Then we can solve the dynamics of vanilla GAN as:

$$\begin{cases} \Phi(s) = \frac{2s}{4s^2+2s+1}C(s), \\ \Theta(s) = \frac{1}{4s^2+2s+1}C(s). \end{cases} \tag{A.14}$$

A.2. Non-saturation GAN

Non-saturation GAN (NS-GAN) shares the same equilibrium point and the objective function for the discriminator. It modifies h_3 as $h_3(x) = \log(\sigma(x))$ and we have:

$$h_3'(x) = (1 - \sigma(x)), h_3''(x) = -\sigma(x)(1 - \sigma(x)). \tag{A.15}$$

By substituting the above equation to $\mathbf{T}$, the dynamic of NS-GAN is equivalent to vanilla GAN and therefore shares the same transfer function.

A.3. Hinge GAN

For Hinge GAN, we have:

$$h_1(x) = \min\{-1 + x, 0\}, \tag{A.16}$$

$$h_2(x) = \min\{-1 - x, 0\}, \tag{A.17}$$

$$h_3(x) = x. \tag{A.18}$$

Then we have:

$$h_1'(x) = 1, h_1''(x) = 0, \tag{A.19}$$

$$h_2'(x) = -1, h_2''(x) = 0, \tag{A.20}$$

$$h_3'(x) = 1. h_3''(x) = 0. \tag{A.21}$$

Therefore the Hinge GAN actually shares the same dynamics as WGAN around the equilibrium point.

A.4. Least Square GAN

The objective function of least square GAN (LS-GAN) is:

$$V_1(\phi;\theta) = -(\phi c - 1)^2 - (\phi\theta)^2, \tag{A.22}$$

$$V_2(\theta;\phi) = -(\phi\theta)^2. \tag{A.23}$$

In this case, there's no equilibrium point. We modify the discriminator as $D(x) = \phi x + 0.5$ which is equilvalent to convert the objective functions as follows:

$$h_1(x) = -(x - 0.5)^2, \tag{A.24}$$

$$h_2(x) = -(x + 0.5)^2, \tag{A.25}$$

$$h_3(x) = -(x - 0.5)^2, \tag{A.26}$$

and therefore we have:

$$h_1'(x) = -2(x - 0.5), h_1''(x) = -2, \tag{A.27}$$

$$h_2'(x) = -2(x + 0.5), h_2''(x) = -2, \tag{A.28}$$

$$h_3'(x) = -2(x - 0.5), h_3''(x) = -2. \tag{A.29}$$

Then the $\mathbf{T}$ can be denoted as:

$$\mathbf{T} = \begin{bmatrix} -4 & -1 \\ 1 & 0 \end{bmatrix}. \tag{A.30}$$

We have:

$$\Phi(s) = -\frac{1}{s+4}(\Theta(s) - C(s)), \tag{A.31}$$

$$\Theta(s) = \frac{1}{s}\Phi(s). \tag{A.32}$$

Then we can solve the dynamics of LSGAN as:

$$\begin{cases} \Phi(s) = \frac{s}{s^2+4s+1}C(s), \\ \Theta(s) = \frac{1}{s^2+4s+1}C(s). \end{cases} \tag{A.33}$$

B. Dynamics with Lipschitz Continuity

In this section, we prove that around the equilibrium, the dynamics of regularized D with Lipschitz constraint is equivalent to the unregularized D as in Eqn. (20). With the dynamics defined by the corresponding gradient flow, we only need to prove that updating D according to Eqn. (20) will not violate the Lipschitz constraints, at least locally around the equilibrium. Here we make the following assumptions:

1. Both $p(x)$ and $p_G(t,x)$ are C^1-smooth: $\frac{dp(x)}{dx}$ and $\frac{dp_G(t,x)}{dx}$ exists and is continuous $\forall\, t$.
2. $q(x) \to 0$ and $\frac{dq(x)}{dx} \to 0$ when $x \to 0$ for $q \in \{p, p_G\}$.
3. There exists an M such that $|\frac{dq(x)}{dx}|_2 < M$ for $q \in \{p, p_G\}$.

The above assumptions are satisfied for most probability density functions.

The distance in the function space is defined as $d(p_1, p_2) = \sup_{x\in\mathbb{R}^n} |p_1(x) - p_2(x)|$ which always exists because of the 2-nd conditions above. We define $\Omega_L = \{p(x) | p(x) \in C^1, |\frac{dp(x)}{dx}|_2 < L\ \forall x.\}$ and $B(\epsilon) = \{p(x) | p(x) \in C^1, \sup_x |p(x)| < \epsilon\}$. Then we have the follow theorem:

Theorem 1. *There exists $\eta > 0$, such that $\forall D(x) \in \Omega_{0.5}$, we have $D(x) + \eta(p(x) - p_G(x)) \in \Omega_1$.*

Proof. By denoting $D'(x) = D(x) + \eta(p(x) - p_G(x))$, We have:

$$\begin{aligned} &\frac{d(D(x) + \eta(p(x) - p_G(x)))}{dx} \\ &= \frac{dD(x)}{dx} + \eta(\frac{p(x)}{dx} - \frac{p_G(x)}{dx}). \end{aligned} \tag{B.1}$$

Therefore, we have

$$|\frac{d(D(x) + \eta(p(x) - p_G(x)))}{dx}|_2 \tag{B.2}$$

$$\leq |\frac{dD(x)}{dx}|_2 + \eta(|\frac{p(x)}{dx}|_2 + |\frac{p_G(x)}{dx})|_2 \tag{B.3}$$

$$\leq 0.5 + \eta(M + M). \tag{B.4}$$

By letting $\eta = \frac{1}{4M}$, we have $|\frac{d(D')}{dx}|_2 \leq 0.75$. Therefore we have $D'(x) \in \Omega_1$. □

The above theorem indicates that when $D(x)$ is sufficient close to the equilibrium and the learning rate is sufficient small, then the dynamics of D still follows Eqn. (11) for Dirac GAN and Eqn. (22) for normal GANs. The simulated results of Dirac GAN in Fig. 1 and the bad performance of SN-GAN with WGAN's objective in Sec. 6.2 agree with this argument.

C. Interpreting CLC in the Parameter Space

In this paper, we mainly analyze our proposed method in the function space, including dynamic analysis and controller designing. Instead, our proposed method can also be interpreted as certain regularization terms on the Jacobian matrix of the training dynamics. Below we provide a formal demonstration.

First, we denote the equilibrium of G and D as (θ^*, ϕ^*), where $p_G(x;\theta^*) = p(x)$ and $D(x;\phi^*) = 0$ for all x. Note that ϕ^* is also a global minimum point of the regularization term $R(D) = \int D^2(x)dx$. Then we have $\frac{\partial^2 R(D)}{\partial\phi^2} \succeq 0$.

We denote $U(D, G)$ as the objective function of the minimax optimization problem in WGAN without CLC regularization. Then the Jacobian matrix of the training dynamic

can be denoted as:

$$J = \begin{pmatrix} \frac{\partial^2 U(D,G)}{\partial \phi^2} & \frac{\partial^2 U(D,G)}{\partial \phi \partial \theta} \\ \frac{\partial^2 U(D,G)}{\partial \theta \partial \phi} & \frac{\partial^2 U(D,G)}{\partial \theta^2} \end{pmatrix}. \tag{C.1}$$

Because of the linearity of the derivation operation, the training dynamics of the WGAN with CLC regularization is denoted as:

$$J' = J - J_L = J - \begin{pmatrix} \frac{\partial^2 L(D)}{\partial \phi^2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{C.2}$$

where we abuse the $\mathbf{0}$ to denote the zero matrix with certain size to match the size of J. Since $\frac{\partial^2 R(D)}{\partial \phi^2} \succeq 0$, we have $-J_L \preceq 0$. Therefore, the CLC regularization introduces a negative semi-definite matrix to the original Jacobian matrix, which is helpful to stabilize the training dynamics of GANs.

D. Understanding Existing Work as Closed-loop Control

A side contribution of this paper is to understand existing methods (Gidel et al., 2018) uniformly as certain CLC controllers. The momentum is an example where Gidel et al. (2018) provide some theoretical analysis of momentum in training GANs. Here we re-analyze the momentum using Dirac GAN under the perspective of control theory.

The momentum method (Qian, 1999) is powerful when training neural networks, whose theoretical formulation is given by:

$$\tilde{\phi}_{t+1} = \beta \tilde{\phi}_t + (1-\beta)\nabla \phi_t,\ \phi_{t+1} = \phi_t + \eta \tilde{\phi}_{t+1}, \tag{D.1}$$

where $\nabla \phi$ is the input of ϕ's dynamic, i.e., $u_D = c - \theta$. The β is the coefficient for the exponential decay. However, momentum instead is not helpful when training GANs (Radford et al., 2015; Mescheder et al., 2018; Brock et al., 2018; Gulrajani et al., 2017) where smaller β or even zero is recommended to achieve better performance.

In control theory, the momentum is equivalent to adding an exponential decay to the input of the dynamics (An et al., 2018):

$$\tilde{h}(t) = \int_0^t h(u) \exp(-\tau(t-u)) du. \tag{D.2}$$

The LT of an exponential decay dynamic is $\frac{1}{s+\tau}$, i.e., $\tilde{H}(s) = \frac{1}{s+\tau} H(s)$. $\tau > 0$ denotes the decay coefficient which depends on β. Therefore, we can formulate the dynamics of Dirac GAN in the following:

$$\begin{cases} m_\phi(t) = \int_0^t (c(u) - \theta(u)) \exp(-\tau(t-u)) du, \\ \frac{d\phi}{dt} = m_\phi(t), \\ \frac{d\theta}{dt} = \phi(t). \end{cases} \tag{D.3}$$

By applying LT, we have $M_\phi(s) = \frac{1}{s+\tau}(C(s) - \Theta(s)$ and Φ can be represented as:

$$\Phi(s) = \frac{s}{s^3 + \tau s^2 + 1} C(s).$$

With a positive τ, there is at least one pole of this dynamic whose real part is larger than 0, indicating the instability of the dynamics for GANs with momentum. The result is consistent with (Gidel et al., 2018).

E. Further Experimental Results on Synthetic Data

In this section, we evaluate our proposed method on a mixture of Gaussian on the two dimensions. The data distribution consists of 8 2D isotropic Gaussian distributions arranged in a ring, where the radius of the ring is 1, and the deviation of each component Gaussian distribution is 0.05. For the coefficient λ, we follow the setting in the spectral normalization as $\lambda \in \{0.01, 0.05, 0.1\}$. We adopt two-layer MLPs for both the generator and the discriminator which consist of $128 - 512$ units. The batch size is is 512.

The generated results are illustrated in Fig. 1 and we further provide the dynamics of the generator distribution in Fig. 2. As we can see, the unregularized WGAN and SGAN suffer from severe model collapse problem and cannot cover the whole data distribution. Besides, the oscillation can be observed during the training process of WGAN: the generator distribution oscillates among the modes of data distribution. Our method can successfully cover all modes compared to the WGAN and SGAN and the dynamics are converged instead of oscillation.

References

An, W., Wang, H., Sun, Q., Xu, J., Dai, Q., and Zhang, L. A pid controller approach for stochastic optimization of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8522–8531, 2018.

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

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.

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.

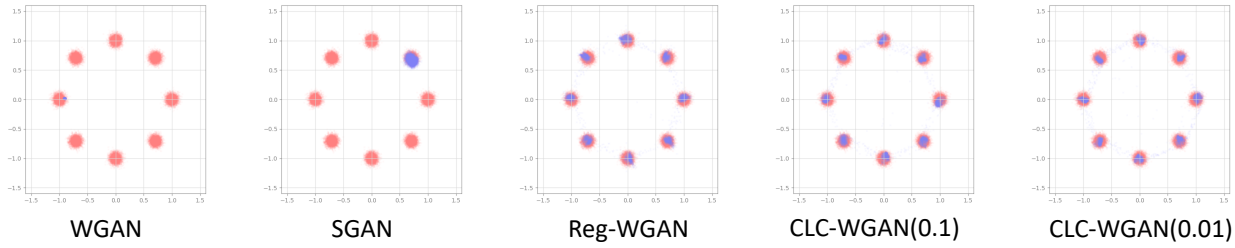

Figure 1: The generated samples for mixture of gaussian distribution. The red points demonstrate the location of data distribution and the blue points are generated samples. Each distribution is plotted using kernel density estimation with 50,000 samples.

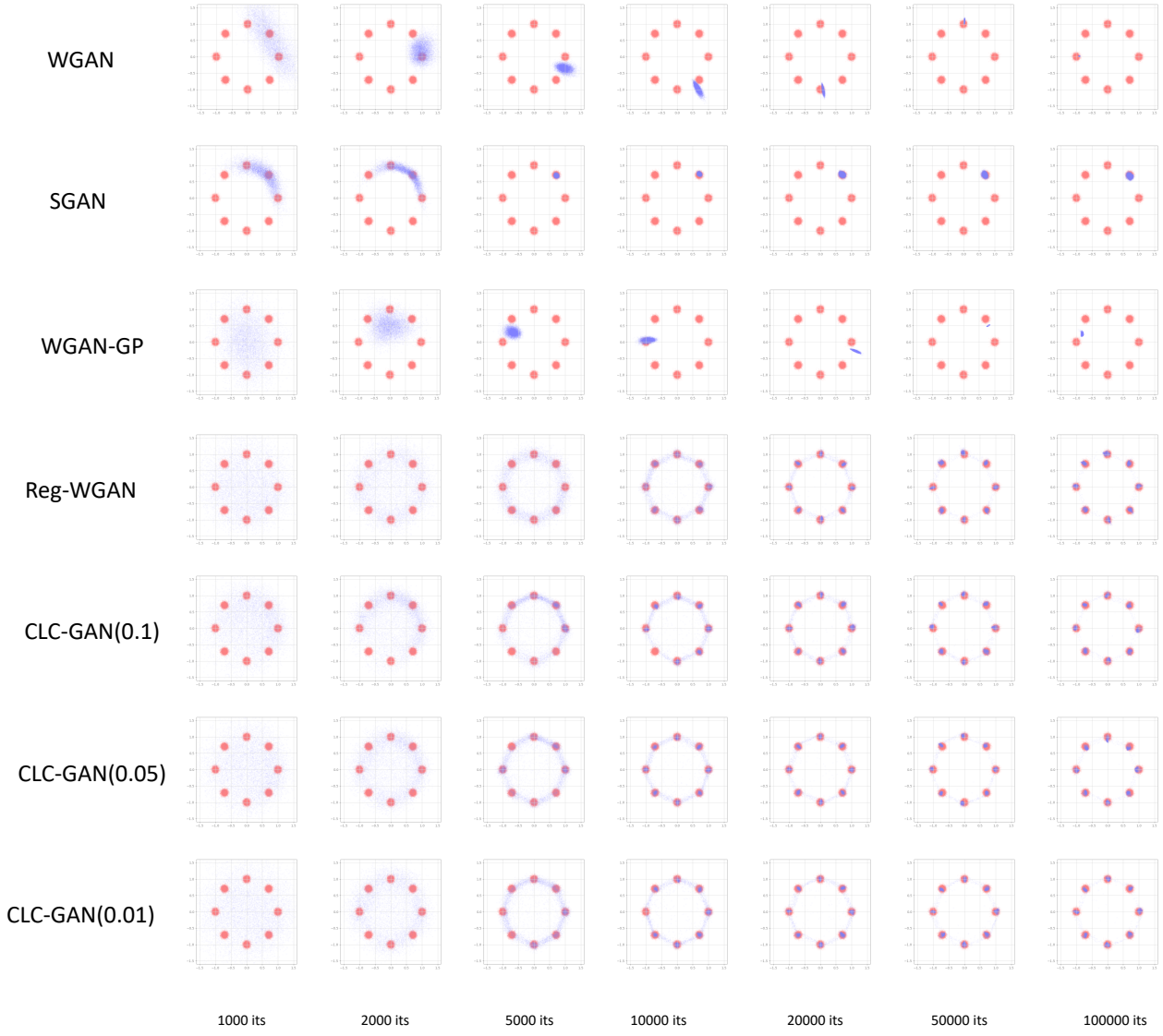

Figure 2: The training dynamics of various GANs on synthetic data.

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.

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., 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.

Qian, N. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

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