Nonstationary GANs: Analysis as Nonautonomous Dynamical Systems

Arash Mehrjou [1] **Bernhard Schölkopf** [1]

Abstract

Generative adversarial networks are used to generate images but still their convergence properties are not well understood. There have been a few studies who intended to investigate the stability properties of GANs as a dynamical system. This paper can be seen in that direction. Among the proposed methods for stabilizing training of GANs, some of them modify the data distribution during the course of training. We unify these methods under the name nonautonomous GAN and investigate their dynamical behaviour when the data distribution is not stationary. We provide theoretical analysis which is supported by simple simulations along with experiments on high dimensional datasets.

1. Introduction

Generative adversarial nets (Goodfellow et al., 2014) are trained by optimizing an objective function over two sets of parameters $\{\theta, \psi\}$. For ease of presentation, the framework is described as a competition between two functions, generator and discriminator who want to minimize/maximize a mutual objective. The GAN objective in its general shape can be written as

$$\arg\min_{\theta}\max_{\psi}\mathcal{L}(\theta,\psi) = \mathop{\mathbb{E}}_{p(z)}[f(D_\psi(G_\theta(z)))] + \quad (1)$$

$$\mathop{\mathbb{E}}_{p_D(x)}[f(-D_\psi(x))] \quad (2)$$

Where ψ parameterizes the discriminator and θ parameterizes the generator. Different choices for $f(.)$ gives various GAN objectives, e.g. Jenson-Shannon (Goodfellow et al., 2014), Wassestein-GAN (Arjovsky et al., 2017), f-GAN (Nowozin et al., 2016), etc. In accordance with these works, we assume $f'(x) \neq 0$. The ultimate goal is to converge to a saddle point where neither discriminator nor generator can achieve a better objective when the other one is kept fixed. Let's call this point in the (θ, ψ) space the *favorite equilibrium*. The interesting property of this point is that $G_\theta(z) = p_D(x)$ in the sense of probability measures. Currently, people are using stochastic gradient descent (SGD) updates to alternately perturb θ and ψ in a hope to converge to the *favorite equilibrium* in the end. Even though the results look visually promising, the dynamical behavior of this system needs more investigation.

In this paper, we restrict ourselves to a minimal example and try to get insight of a GAN whose target distribution is not fixed. This strategy has been employed in a couple of works (Arjovsky & Bottou, 2017; Sønderby et al., 2016). Especially in β-GAN (Mehrjou et al., 2017), authors proposed an annealing strategy that heats up data to a high entropy uniform distribution. Once uniform distribution is learned by the generator, the target distribution is cooled towards the original data distribution. The main focus of the current paper is studying the dynamical behaviour of GAN when the target distribution is also dynamic. This means that the environment of the GAN components (Generator + Discriminator) which is the data distribution is time variant. Borrowing the terminologies from dynamical systems (Khalil, 1996), we call this setting, *nonautonomous* GAN.

2. Nonautonomous GAN

Continuous dynamical system— We see GAN as a continuous dynamical system. This assumption is valid when the learning rate of Stochastic Gradient Descent (SGD) tends to zero in optimization, i.e. $\epsilon \to 0$.

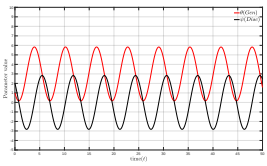
Autonomous GAN —In conventional GAN training, the dynamical system

$$\begin{cases} \dot{\theta} = -\nabla_\theta \mathcal{L}(\theta,\psi) \\ \dot{\psi} = \nabla_\psi \mathcal{L}(\theta,\psi) \end{cases} \quad (3)$$

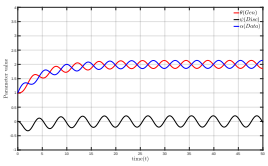
is an approximation of the training pattern for a tiny learning rate. We call these dynamical systems *autonomous* because the right-hand side function is not an explicit function of time (Khalil, 1996). Given the Lipschitz continuity of the right-hand side of Eq. 3, there exists a solution for this system and it is unique.

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany. Correspondence to: Arash Mehrjou <arash.mehrjou@tuebingen.mpg.de>.

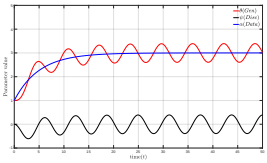
Presented at the ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models



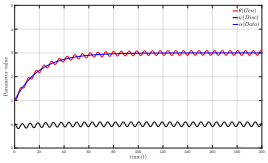
(a) Autonomous GAN



(b) β'-GAN: coupled-dynamics



(c) β-GAN: decoupled-dynamics



(d) β-GAN: decoupled-dynamics (long T)

Figure 1. State evolution of various continuous dynamical systems approximating the behavior of GANs when the learning rate is small $\epsilon \to 0$.(a) Normal GAN: Two-state autonomous system with static data distribution. (b) β'-GAN: Three-state nonautonomous system when the dynamics of data distribution is coupled with the other states. (c) β-GAN: Three-state nonautonomous system when the dynamics of data distribution is only governed by the annealing process. (d) The same as (c) but with slower annealing process.

Nonautonomous GAN— The overall idea is introducing a new state α in the GAN objective function in Eq. 2. This state controls the data distribution. More precisely, the objective function becomes

$$\mathcal{L}(\theta, \psi, \alpha) = \mathop{\mathbb{E}}_{p(z)}[f(D_\psi(G_\theta(z)))] + \mathop{\mathbb{E}}_{p_D(x;\alpha)}[f(-D_\psi(x))] \tag{4}$$

To study the effect of this new state, we introduce a minimalistic framework called *tiny*-GAN to analytically compute the solution.

tiny-GAN —To have a minimal tractable GAN framework, we set $P_D(x;\alpha) = \delta_\alpha$ and $G_\theta(z) = \delta_\theta$, meaning that, the real data is concentrated on a single point at $x = \alpha$ and the generator is only capable of generating one point at location $x = \theta$. The discriminator is assumed linear, i.e. $D_\psi(x) = \psi x$. In contrast to (Mescheder, 2018), we do not tie data to the origin and release it to occupy any location on the real axis. After these simplifications, the objective function of Eq. 2 becomes:

$$\mathcal{L}(\theta, \psi, \alpha) = f(\psi\theta) + f(-\psi\alpha) \tag{5}$$

and the dynamical system of training GAN in Eq. 3 is written as

$$\begin{cases} \dot{\theta} = -\psi f'(\psi\theta) \\ \dot{\psi} = \theta f'(\psi\theta) - \alpha_r f'(-\psi\alpha_r) \end{cases} \tag{6}$$

In this formulation, α_r is fixed and represents real data distribution.

Many formulations of GAN can be characterized by the dynamical system of Eq. 3 which contains only two states: the parameters of the generator (θ) and the parameters of the discriminator (ψ). Here we augment the state-space equation with a new state which characterizes the properties of the data distribution $p_D(x;\alpha)$. In harmony with the minimalistic nature of *tiny*-GAN, the entire data distribution is characterized by α here. Notice that the real data distribution is not dynamic. Indeed, real data distribution is the target point of the dynamics of $\alpha(t)$ and we represent it by α_r, i.e. $\alpha(t) \to \alpha_r$ as $t \to \infty$. Optimizing Eq. 4 when the dynamics of α is only governed by $\nabla_\alpha \mathcal{L}(\theta, \psi, \alpha)$ results in trivial answers since there will be no motivation that $\alpha(t)$ and $\theta(t)$ arrives at *favorite equilibrium* where $\alpha(\infty) = \alpha_r$. To cure this issue, β-GAN suggested a full annealing strategy over $p_D(x;\alpha)$. This idea turns the dynamical system of Eq. 3 into a time-varying (Nonautonomous) system. At this point two branches can be thought of. In the first branch which is also the method devised by β-GAN, α has partially decoupled dynamics from the other states of the system. By partially decoupled, we mean that the dynamics of α is not affected by the dynamics of the other states of the system. However, the dynamics of the other states may depend on the dynamics of α. In the second branch (let's call it β'-GAN), α undergoes two dynamics. One is the dynamics imposed by the GAN objective $\nabla_\alpha \mathcal{L}(\theta, \psi, \alpha)$ which acts by SGD updates and the other one is the annealing dynamics. The first term makes the dynamics of α coupled with the other states of the system. As proposed in β-GAN, annealing steps must act with a slower timescale than SGD iterations of the optimization. The slow partially decoupled dynamics of α is characterized by

$$\alpha(t) = (\alpha_0 - \alpha_r)e^{-\frac{t}{T}} + \alpha_r \tag{7}$$

where $T > 1$ is a time constant that makes this dynamic term slower than the SGD dynamics. In addition, α_0 is the initial value of $\alpha(t)$ that characterizes the initial distribution of data $p_D(x;\alpha = \alpha_0)$ when the annealing process starts. The parameter α_r is the target value of $\alpha(t)$ for which $p_D(x;\alpha = \alpha_r)$ becomes the real data distribution $p_D(x)$. Therefore the state-space equation is written as follows:

$$\begin{cases} \dot{\theta} = -\psi f'(\psi\theta) \\ \dot{\psi} = \theta f'(\psi\theta) - \alpha f'(-\psi\alpha) \\ \dot{\alpha} = \lambda[-\psi f'(-\psi\alpha)] + \frac{1}{T}(\alpha_r - \alpha_0)e^{-\frac{t}{T}} \end{cases} \tag{8}$$

The hyper-parameter $\lambda \in \{0, 1\}$ is a switch and has an important meaning which differentiate between β-GAN and β'-GAN. When $\lambda = 0$ (β-GAN) the variable α is not perturbed by short timescale SGD updates. This means that α has partially decoupled dynamics from the dynamics of states $\{\theta, \psi\}$. On the other hand, when $\lambda = 1$, the dynamics of α is governed by both a short timescale term and a long timescale term. The former is the SGD updates

and the latter is the same as in β-GAN. Furthermore, β-GAN suggests starting from *uniform* distribution meaning that $p_D(x;\alpha)$ is constant over a specified area and zero elsewhere. In β-GAN, the generator must be pre-trained to capture the uniform distribution for a certain data dimension n. This means that the generator at time $t = 0$ is able to generate a simple uniform distribution which matches the initial distribution $p_D(x;\alpha = \alpha_0)$. In our minimalistic setting of *tiny*-GAN and the dynamical system of Eq.8, this translates to $\theta(0) = \alpha(0) = \alpha_0$.

3. Simulations

To show the effect of annealing strategy in GANs, simple simulations are presented here for autonomous GAN, β-GAN and β'-GAN. Note that the objective function of Eq. 2 becomes that of W-GAN when $f(y) = y$ (Arjovsky et al., 2017). We compare normal (autonomous) GAN with two Nonautonomous GANs (β-GAN and β'-GAN). Remember that in β-GAN, data distribution does not change with short timescale and it has its own partially decoupled dynamics due to annealing while in β'-GAN, data distribution is altered by both the fast dynamics of SGD and the slow dynamics of annealing. In all simulated experiments, the real data distribution is located at $x = \alpha_r = 3$ which is the static value of α for autonomous GAN but target value of $\alpha(t)$ for Nonautonomous GANs. Fig. 1(a) shows the solution of the dynamical system of Eq. 6 when $f(y) = y$ as in Wasserstein GAN with initial point $(\theta(0), \psi(0)) = (1, 2)$. As can be seen, the states (θ, ψ) are oscillating around (θ^*, ψ^*)=$(3, 0)$ which is the equilibrium point of this system. For the linear $f(y) = y$ and *tiny*-GAN frameworks which are studied in this paper, this result is global. It can be shown that for nonlinear choices of $f(y)$, the same oscillation is observable but locally around the equilibrium point. Notice that this oscillation is so called *unsustained oscillation* which is different from stable limit cycles (Khalil, 1996; Isidori, 2013). Here, the amplitude of the oscillation depends on the initial state $(\theta(0), \psi(0))$ which is an undesirable effect. Fig. 1(b) depicts the behavior of β'-GAN and shows the solution to the dynamical system of Eq. 8 when $\lambda = 1$ with initial states $(\theta(0), \psi(0), \alpha(0)) = (1, 0, 1)$. Still the target value for $\alpha(t)$ is $\alpha_r = 3$. As can be seen, the dynamical system is still oscillating but the amplitude of the oscillation is reduced. The undesirable point of this setting is that the system is now oscillating around a wrong equilibrium point $(\theta, \psi, \alpha) = (2, 0, 2)$ which is different from the *favorite equilibrium* $(\theta, \psi, \alpha) = (3, 0, 3)$.

Fig. 1(c) simulates the behavior of β-GAN by running the dynamical system of Eq. 8 with $\lambda = 0$ from the initial states $(\theta(0), \psi(0), \alpha(0)) = (1, 0, 1)$. Again $T = 3$ and the target data is $\alpha_r = 3$. As can be seen, the system is oscillating as Fig. 1(b) but this time around the correct point $(\theta, \psi, \alpha) = (3, 0, 3)$. The amplitude of oscillation is lower than autonomous GAN of Fig. 1(a) and decreases more by increasing T. Increasing T means it takes longer for α to move from α_0 to α_r which is equivalent to slower annealing dynamics or finer annealing steps in discrete setting. This is shown in Fig. 1(d) where the entire setting is as the previous case but $T = 30$ results in slower approach to a_r but reduced oscillation amplitude around the correct equilibrium point. These empirical observations are proved in the next section by solving GAN equations for an analytic solution.

4. Theoretical Analysis

The simulations of section 3 shows that the amplitude of oscillation decreases as T increases in β-GAN framework. Here, a more formal analysis is provided to explain this observation. The dynamical system of Eq. 8 for $f(y) = y$ and $\lambda = 0$ will be written as follows:

$$\begin{cases} \dot{\theta} &= -\psi \\ \dot{\psi} &= \theta - \alpha \\ \dot{\alpha} &= \frac{1}{T}(\alpha_r - \alpha_0)e^{-\frac{t}{T}}. \end{cases} \tag{9}$$

Let's $a = \frac{1}{T}$ and $K = \frac{(\alpha_r - \alpha(0))}{T}$. We take Laplace transform from both sides of three equations above:

$$\begin{cases} s\theta(s) - \theta(0) &= -\psi(s) \\ s\psi(s) - \psi(0) &= \theta(s) - \alpha(s) \\ s\alpha(s) - \alpha(0) &= \frac{K}{s+a}. \end{cases} \tag{10}$$

Taking derivative of the both sides of the second line of Eq. 9 amounts to multiplying both sides of the second line of Eq. 10 by Laplace differentiation operator s and results in

$$s^2\psi(s) = s\theta(s) - s\alpha(s) = \cancel{\theta(0)} - \psi(s) - \cancel{\alpha(0)} - \frac{K}{s+a} \tag{11}$$

where $\theta(0)$ and $\alpha(0)$ cancels each other due to the assumption of β-GAN that generator starts from a simple initial distribution characterized by $\alpha(0)$. This assumption consequently ensures $\psi(0) = 0$ because it is assumed that the equilibrium is initially found for both generator and discriminator for the data distribution $\alpha(0)$. Solving for $\psi(s)$ gives us

$$\psi(s) = \frac{-K}{(1+s^2)(s+a)}. \tag{12}$$

We then expand the right-hand side as a sum of polynomial fractions:

$$\psi(s) = \frac{-K}{1+a^2}\frac{s}{1+s^2} + \frac{Ka}{1+a^2}\frac{1}{1+s^2} + \frac{K}{1+a^2}\frac{1}{s+a}. \tag{13}$$

Computing inverse Laplace transform of $\psi(s)$ gives

$$\mathcal{L}^{-1}\{\psi(s)\} = \overbrace{A\cos(t) + B\sin(t)}^{\psi_1(t)} + Ce^{-at}. \tag{14}$$

where $A = \frac{-K}{1+a^2}$, $B = \frac{Ka}{1+a^2}$, and $C = \frac{K}{1+a^2}$. The last term vanishes in the steady state solution when $t \to \infty$. We are mainly interested in the first two parts which are responsible for the persistent oscillation. Adding two harmonics results in a new harmonic with scaled amplitude $\mathcal{A}$ and phase shift ϕ:

$$\begin{cases} \psi_1(t) = \mathcal{A}\sin(t+\phi) \\ \mathcal{A} = \sqrt{A^2 + B^2 + 2AB\cos(\pi/2)} \\ \phi = \tan^{-1}(A, B) \end{cases} \tag{15}$$

where $\tan^{-1}$ is quadrant-aware arc tangent. By substituting A and B in $\mathcal{A}$ we can compute the amplitude of the persistent oscillation as

$$\mathcal{A} = \sqrt{\left(\frac{K}{1+a^2}\right)^2 + \left(\frac{Ka}{1+a^2}\right)^2} = \frac{K}{1+a^2}\sqrt{(1+a^2)}. \tag{16}$$

The term $1 + a^2 = 1 + \frac{1}{T^2} \to 1$ as $T \to \infty$. The important term is K that goes to zero as $T \to \infty$ and proves our claim (rooted in observations such as Fig. 1) that the oscillation amplitude decreases as the annealing time T increases. Now that the analytic form of $\psi(t)$ is known, we can move on and obtain the analytic form of $\theta(t)$. According to Eq.10, we can write $\theta(s)$ in terms of $\psi(s)$ as:

$$\theta(s) = \frac{1}{s}[\theta(0) - \psi(s)] \tag{17}$$

where $\frac{1}{s}$ acts as an integrator. Therefore, we can obtain the inverse Laplace transform $\mathcal{L}^{-1}\{\theta(s)\}$ and compute the following definite integral to compute $\theta(t)$ as

$$\begin{aligned} \theta(t) &= \theta(0)\mathcal{L}^{-1}\{\frac{1}{s}\} - \int_{\tau=0}^{\tau=t} \psi(\tau)\,\mathrm{d}\tau \\ &= \theta(0) - \frac{K}{1+a^2}\int_{\tau=0}^{\tau=t} e^{-at}\,\mathrm{d}\tau + \int_{\tau=0}^{\tau=t}\psi_1(\tau)\,\mathrm{d}\tau \\ &= \theta(0) + \frac{K}{1+a^2}\frac{1}{a} + \int_{\tau=0}^{\tau=t}\psi_1(\tau)\,\mathrm{d}\tau \\ &= \theta(0) + \frac{\alpha_r - \alpha_0}{1+a^2} + \underbrace{\int_{\tau=0}^{\tau=t}\psi_1(\tau)\,\mathrm{d}\tau}_{\Psi_1(t)}. \end{aligned} \tag{18}$$

Notice that $\Psi_1(t)$ is the integral of a sinusoidal which is itself a sinusoidal. As the annealing time increases, $T \to \infty$, the term $1 + a^2 = 1 + \frac{1}{T^2} \to 1$ and we eventually have the steady state solution of $\theta(t)$ as follows:

$$\lim_{T\to\infty} \theta(t) = a_r + \Psi_1(t) \tag{19}$$

which shows the persistent oscillation around the desired equilibrium point a_r that is the real data distribution. □

5. High dimensional example

The goal of the current paper is studying the effect of annealing in adversarial training. Hence, the goal of this section is not comparing performance of nonautonomous GANs with each other or other stabilizing strategies. Instead, we want to argue that what we have shown in section 3 and proved in section 4 can be observed in more realistic scenarios. This section shows the practical results of one nonautonomous GAN algorithm named β-GAN (Mehrjou et al., 2017). Here, data distribution is controlled by a parameter called β (that mimics the role of α in Eq. 8). The abstract idea is categorized under the title nonautonomous GANs that was introduced earlier in this paper. However, to be self-contained, some of the implementation detail of the algorithm comes in the following.

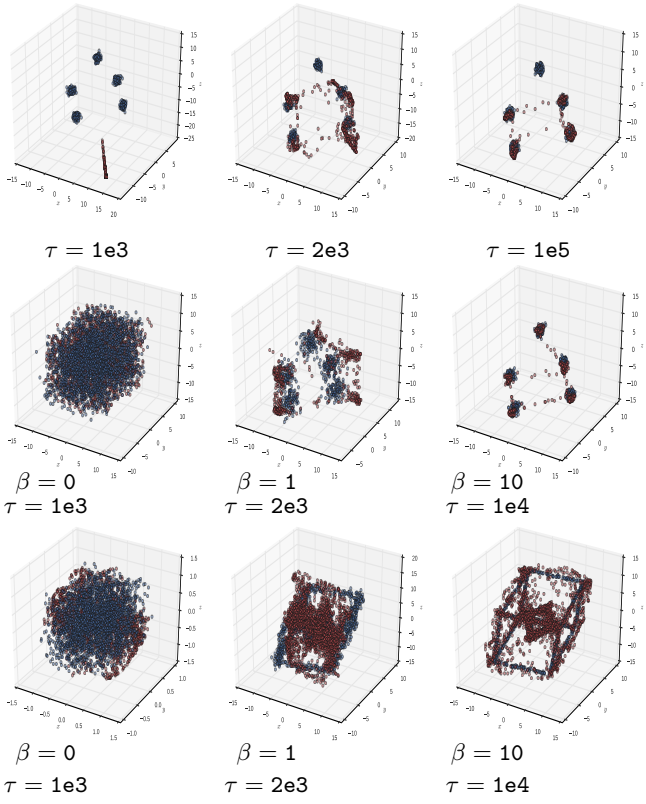
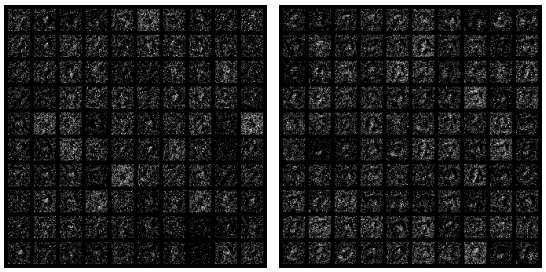


Figure 2. Three dimensional example — Top row: The performance of vanilla GAN on a mixture of five Gaussian components in three dimensions. *Middle row:* The performance of β-GAN on the same dataset. *Bottom row:* The performance of β-GAN on the synthesized mixture of two cubes. Blue/red dots are real/generated data. To compare the computational cost, we report τ, which is the total number of gradient evaluations from the start. We use the architecture G:[z(3) | ReLU(128) | ReLU(128) | Linear(3)] and D:[x(3) | Tanh(128) | Tanh(128) | Tanh(128) | Sigmoid(1)] for generator and discriminator where the numbers in the parentheses show the number of units in each layer. The annealing parameters are $[\beta_1 = 0.1,\ \beta_K = 10,\ K = 20]$.

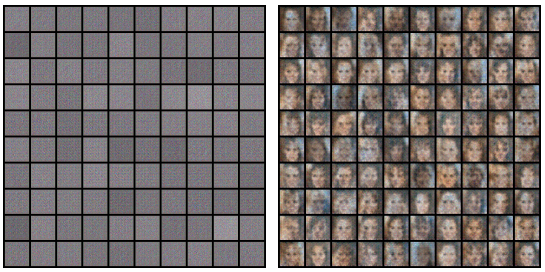
We assume the generative and discriminative networks $\mathbb{G}$ and $\mathbb{D}$ have very large capacity, parameterized by deep neu-



(a) Generated samples for $\beta = 0.1$ (b) Generated samples for $\beta = 1$

(c) Generated samples for $\beta = 5$ (d) Generated samples for $\beta = \infty$

Figure 3. β-GAN trained on MNIST with $\dim(z) = 28 \times 28$ — Samples generated from MNIST during annealing procedure. The network starts from generating the uniform distribution $\mathrm{Uniform}[-1, 1]^{28\times28}$ and gradually generates samples corresponding to each value of β. We use the fullly connected architecture G:[z(784) | BNReLU(256) | BNReLU(256) | BNReLU(256) | Linear(784)] and D:[x(784) | BNReLU(256) | BNReLU(512) | BNReLU(512) | Sigmoid(1)] for generator and discriminator where the numbers in the parentheses show the number of units in each layer. BNReLU is batch normalization (Ioffe & Szegedy, 2015) concatenated with ReLU activation. The annealing parameters are $[\beta_1 = 0.1,\ \beta_K = 10,\ K = 20]$ the same as 3D experiment in Fig. 2.



(a) Generated samples for $\beta = 0.1$ (b) Generated samples for $\beta = 1$

(c) Generated samples for $\beta = 5$ (d) Generated samples for $\beta = \infty$

Figure 4. β-GAN trained on CelebA with $\dim(z) = 64 \times 64 \times 3$ — Samples generated from CelebA dataset during annealing procedure. The network starts from generating the uniform distribution $\mathrm{Uniform}[-1, 1]^{64\times64\times3}$ and gradually generates samples corresponding to each value of β. We borrowed DCGAN architecture from (Radford et al., 2015) except that the input noise of the generative network has the dimension of data and the output layer is changed to linear instead of Tanh. The annealing parameters are $[\beta_1 = 0.1,\ \beta_K = 10,\ K = 20]$ the same as 3D experiment in Fig. 2.

ral networks $G(z;\theta_G)$ and $D(x;\theta_D)$. Here, $z \sim p(z)$ is the (noise) input to the generative network $G(z;\theta_G)$, and $D(x;\theta_D)$ is the discriminative network that is performing logistic regression. The discriminative network is trained with the binary classification labels $D = 1$ for the N observations $\{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\} \in \mathcal{R}^d$, and $D = 0$ otherwise. The GAN objective is to find θ^*_G such that $G(z;\theta^*_G) \sim p_{\mathrm{data}}(x)$. This is achieved at the Nash equilibrium (favorite equilibrium) of the following minimax objective:

$$\begin{aligned}\theta^*_G &= \arg\min_{\theta_G}\max_{\theta_D} f(\theta_D,\theta_G),\\ f(\theta_D,\theta_G) &= \mathop{\mathbb{E}}_{x\sim p_{\mathrm{data}}} \log\left(D(x;\theta_D)\right) + \\ &\mathop{\mathbb{E}}_{z\sim p(z)} \log(1 - D(G(z;\theta_G);\theta_D)),\end{aligned} \tag{20}$$

where at the equilibrium $D(G(z;\theta^*_G);\theta^*_D) = 1/2$ (Goodfellow et al., 2014). One way to introduce β is to go back to the empirical distribution and rewrite it as a mixture of Gaussians with zero widths:

$$\begin{aligned}p_{\mathrm{data}}(x) &= \frac{1}{N}\sum_i \delta(x - x^{(i)})\\ &= \frac{1}{N}\lim_{\beta\to\infty}\sqrt{\frac{\beta}{2\pi}}\sum_i \exp\left(-\frac{\beta(x-x^{(i)})^2}{2}\right).\end{aligned}$$

The heated data distribution at finite β is therefore given by:

$$p_{\mathrm{data}}(x;\beta) = \frac{1}{N}\left(\frac{\beta}{2\pi}\right)^{d/2}\sum_i \exp\left(-\frac{\beta(x-x^{(i)})^2}{2}\right).$$

The d-dimensional box— The starting point in β-GAN is to learn to sample from the uniform distribution. Since the uniform distribution is not normalized in $\mathcal{R}^d$, we set $\mathcal{X}$ to be the finite interval $[a, b]^d$. The uniform distribution sets the scale in our framework, and the samples $x_\beta \sim p_{\mathrm{data}}(x;\beta)$ are rescaled to the same interval. This hard d-dimensional "box" for the data "particles" is thus assumed throughout the

Algorithm 1 Minibatch stochastic gradient descent training of annealed generative adversarial networks. The inner loop can be replaced with other GAN architectures and/or other divergence measures. The one below uses the Jensen-Shannon formulation of Goodfellow *et al.* as the objective, as are other experiments in this paper.

- Train GAN to generate uniform distribution and obtain $\theta^*_{g,0}$ and $\theta^*_{d,0}$.
- Receive β_1, β_K, and K, where K is the number of cooling steps between/including β_1 and β_K.
- Compute $\alpha > 1$ as the geometric cooling factor:

$$\alpha = \left(\frac{\beta_K}{\beta_1}\right)^{\frac{1}{K}}$$

- Initialize β: $\beta \leftarrow \beta_1$
- Initilize $\theta_{g,\beta} \leftarrow \theta^*_{g,0}$ and $\theta_{d,\beta} \leftarrow \theta^*_{d,0}$

for number of cooling steps (K) **do**

  for number of training steps (n) **do**

  - Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p(z)$.
  - Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x; \beta)$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_{d,\beta}} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}; \theta_{d,\beta}\right) + \log\left(1 - D\left(G\left(z^{(i)}; \theta_{g,\beta}\right); \theta_{d,\beta}\right)\right)\right].$$

  - Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p(z)$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_{g,\beta}} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}; \theta_{g,\beta}\right); \theta_{d,\beta}\right)\right).$$

  end for

  - Increase β geometrically: $\beta \leftarrow \beta * \alpha$

end for

- Switch from $p_{data}(x; \beta_K)$ to the empirical distribution ($\beta = \infty$) for the final epochs.

paper. Its presence is conceptually equivalent to a diffusion process of the data particles in the box $[a, b]^d$, where they diffuse to the uniform distribution like ink dropped in water (Sohl-Dickstein et al., 2015). Here, the distribution is parameterized with β instead of the diffusion time. We also mention a non-Gaussian path to the uniform distribution in the discussion section.

With this setup, the minimax optimization task at each β is:

$$\begin{aligned} \theta^*_G(\beta) = &\quad \arg\min_{\theta_G} \max_{\theta_D} f(\theta_D, \theta_G; \beta), \\ f(\theta_D, \theta_G; \beta) = &\quad \mathbb{E}_{x \sim p_{\text{data}}(x;\beta)} \log\left(D(x; \theta_D)\right) + \\ &\quad \mathbb{E}_{z \sim p(z)} \log(1 - D(G(z; \theta_G); \theta_D)). \end{aligned} \tag{21}$$

Note that the optimal parameters θ^*_G and θ^*_D depend on β implicitly. In β-GAN, the first task is to learn to sample from the uniform distribution. It is then trained simultaneously as the uniform distribution is smoothly annealed to the empirical distribution by increasing β. We chose a simple fixed geometric scheduling for annealing in this work. The procedure is given in Algorithm 1.

The convergence of the algorithm is based on the following conjecture:

In the continuous annealing limit from the uniform distribution to the data distribution GAN remains stable at the equilibrium, assuming G and D have large capacity and that they are initialized at the minimax equilibrium for generating the uniform distribution[1] *in the ambient space* $\mathcal{X}$.

This conjecture is proved for a simplistic setting in section 4. Implemented algorithm is then tested on various datasets whose results are depicted in Fig. 2 (synthetic data), Fig. 3 (MNIST), and Fig.4 (CelebA). The annealing strategy results in better stability in the sense that the generator passes a path towards the target distribution with ever increasing quality of generated images, without forgetting the learned information abruptly , and with less amplitude of oscillation around the equilibrium point. Investigating oscillations in the parameter space as discussed in section. 4 and depicted

[1] This requires $\dim(z) \geq d$.

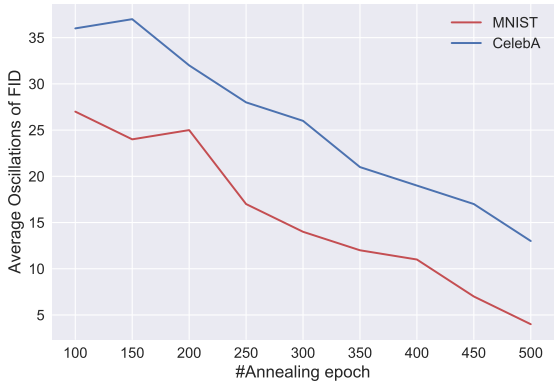


Figure 5. Average ampliude of oscillations of FID around the equilibrium point for different numbers of annealing steps. When the annealing schedule is finer (higher number of annealing steps), the FID shows oscilations with lower amplitude.

in Fig. 3 for *tiny*-GAN is not easily possible in higher dimensions for complex discriminators and generators. However, we used an indirect strategy to observe this effect. We took FID score (Heusel et al., 2017) as a performance metric for the generator and monitored its value over the course of training. FID is computed between two batches of samples generated from $P_D(x;\alpha)$ and $P_{\text{data}}(x)$. We computed the FID score for every 100 iterations that gives a highly noisy curve. To extract the trend, we passed the FID curve through a low-pass filter. We observed the resultant curves and found the knee point after which the oscillations occur on a horizontal line. For a period of $50k$ iterations in this region we computed the average max-to-min amplitude. This procedure was repeated for different number of annealing steps (higher number of steps amounts to finer annealing schedule) and the result is depicted in Fig. 5. This result indirectly confirms our theoretical analysis in the simplistic scenario of section 4 that longer period of annealing results in oscillations with less amplitude. However, since oscillation in the parameter space is not tractably measurable for millions of parameters, we looked at the oscillations in the FID score that indirectly shows an oscillating behaviour in the generator.

6. Conclusion

This paper uses simulations and theoretical analysis to study annealing as a promising approach in GANs. The dynamics of two frameworks called β-GAN and β'-GAN are studied and showed that while annealing with partially decoupled dynamics (β-GAN) results in reduced oscillation, annealing with coupled dynamics can result in convergence to a wrong equilibrium. A minimalistic nonautonomous adversarial dynamical system called *tiny*-GAN is proposed to mimic the behavior of GAN in a tractable way when its data distribution is nonstationary. The optimization updates and the dynamics of the annealing strategy is approximated by a continuous dynamical system. We believe viewing adversarial strategies as dynamical systems are interesting not only in unsupervised learning, but also in control theory where compelling systems may arise when states act in an adversarial way.

Acknowledgements

AM acknowledges comments by Saeed Saremi on the manuscript and valuable ideas on the body of the work.

References

Arjovsky, M. and Bottou, L. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

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.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

Ioffe, S. and Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

Isidori, A. *Nonlinear control systems*. Springer Science & Business Media, 2013.

Khalil, H. K. Noninear systems. *Prentice-Hall, New Jersey*, 2(5):5–1, 1996.

Mehrjou, A., Schölkopf, B., and Saremi, S. Annealed generative adversarial networks. *arXiv preprint arXiv:1705.07505*, 2017.

Mescheder, L. On the convergence properties of gan training. *arXiv preprint arXiv:1801.04406*, 2018.

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

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

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015.

Sønderby, C. K., Caballero, J., Theis, L., Shi, W., and Huszár, F. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.