

---

# Discretization Drift in Two-Player Games

---

Mihaela Rosca<sup>1,2</sup> Yan Wu<sup>1</sup> Benoit Dherin<sup>3</sup> David G.T. Barrett<sup>1</sup>

## Abstract

Gradient-based methods for two-player games produce rich dynamics that can solve challenging problems, yet can be difficult to stabilize and understand. Part of this complexity originates from the discrete update steps given by simultaneous or alternating gradient descent, which causes each player to drift away from the continuous gradient flow — a phenomenon we call discretization drift. Using backward error analysis, we derive modified continuous dynamical systems that closely follow the discrete dynamics. These modified dynamics provide an insight into the notorious challenges associated with zero-sum games, including Generative Adversarial Networks. In particular, we identify distinct components of the discretization drift that can alter performance and in some cases destabilize the game. Finally, quantifying discretization drift allows us to identify regularizers that explicitly cancel harmful forms of drift or strengthen beneficial forms of drift, and thus improve performance of GAN training.

## 1. Introduction

The fusion of deep learning with two-player games has produced a wealth of breakthroughs in recent years, from Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) through to model-based reinforcement learning (Sutton & Barto, 2018; Rajeswaran et al., 2020). Gradient descent methods are widely used across these settings, partly because these algorithms scale well to high-dimensional models and large datasets. However, much of the recent progress in our theoretical understanding of two-player differentiable games builds upon the analysis of continuous differential equations that model the dynamics of training (Singh et al., 2000; Heusel et al., 2017; Nagarajan & Kolter, 2017), leading to a gap between theory and practice. Our aim is to take a step forward in our understanding

of two player games by finding continuous systems which better match the gradient descent updates used in practice.

Our work builds upon Barrett & Dherin (2021), who use backward error analysis to quantify the discretization drift induced by using gradient descent in supervised learning. We extend their work and use backward error analysis to understand the impact of discretization in the training dynamics of two-player games. More specifically, we quantify the *Discretization Drift* (DD), the difference between the solutions of the original ODEs defining the game and the discrete steps of the numerical scheme used to approximate them. To do so, we construct modified continuous systems that closely follow the discrete updates. While in supervised learning DD has a beneficial regularization effect (Barrett & Dherin, 2021), we find that the interaction between players in DD can have a destabilizing effect in adversarial games.

**Contributions:** Our primary contribution, Theorems 3.1 and 3.2, provides the continuous modified systems which quantify the discretization drift in simultaneous and alternating gradient descent for general two-player differentiable games. Both theorems are novel in their scope and generality, as well as their application toward understanding the effect of the discretization drift in two-player games parametrized with neural networks. Theorems 3.1 and 3.2 allow us to use dynamical system analysis to describe GD without ignoring discretization drift, which we then use to:

- Provide new stability analysis tools (Section 4).
- Motivate explicit regularization methods which drastically improve the performance of simultaneous gradient descent in GAN training (Section 7.3).
- Pinpoint optimal regularization coefficients and shed new light on existing explicit regularizers (Table 1).
- Pinpoint the best performing learning rate ratios for alternating updates (Sections 5 and 7.2).
- Explain previously observed but unexplained phenomena such as the difference in performance and stability between alternating and simultaneous updates in GAN training (Section 6).

---

<sup>1</sup>DeepMind, London, UK <sup>2</sup>Center for Artificial Intelligence, University College London <sup>3</sup>Google, Dublin, Ireland. Correspondence to: Mihaela Rosca <mihaelacr@deepmind.com>.

## 2. Background

**Backward Error Analysis:** Backward error analysis was devised to study the long-term error incurred by following an ODE numerical solver instead of an exact ODE solution (Hairer & Lubich, 1997; Hairer et al., 2006). The general idea is to find a modified version of the original ODE that follows the steps of the numerical solver exactly. Recently, Barrett & Dherin (2021) used this technique to uncover a form of DD, called *implicit gradient regularization*, arising in supervised learning for models trained with gradient descent. They showed that for a model with parameters  $\theta$  and loss  $L(\theta)$  optimized with gradient descent  $\theta_t = \theta_{t-1} - h\nabla_{\theta}L(\theta)$ , the first order modified equation is  $\dot{\theta} = -\nabla_{\theta}\tilde{L}(\theta)$ , with modified loss

$$\tilde{L}(\theta) = L(\theta) + \frac{h}{4} \|\nabla_{\theta}L(\theta)\|^2 \quad (1)$$

This shows that there is a hidden implicit regularization effect, dependent on learning rate  $h$  that biases learning toward flat regions, where test errors are typically smaller (Barrett & Dherin, 2021).

**Two-player games:** A well developed strategy for understanding two-player games in gradient-based learning is to analyze the continuous dynamics of the game (Singh et al., 2000; Heusel et al., 2017). Tools from dynamical systems theory have been used to explain convergence behaviors and to improve training using modifications of learning objectives or learning rules (Nagarajan & Kolter, 2017; Balduzzi et al., 2018; Mazumdar et al., 2019). Many of the insights from continuous two-player games apply to games that are trained with discrete updates. However, discrepancies are often observed between the continuous analysis and discrete training (Mescheder et al., 2018). In this work, we extend backward error analysis from the supervised learning setting – a special case of a one-player game – to the more general two-player game setting, thereby bridging this gap between the continuous systems that are often analyzed in theory and the discrete numerical methods used in practice.

## 3. Discretization drift

Throughout the paper we will denote by  $\phi \in \mathbb{R}^m$  and  $\theta \in \mathbb{R}^n$  the row vectors representing the parameters of the first and second player, respectively. The players update functions will be denoted correspondingly by  $f(\phi, \theta) : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^m$  and by  $g(\phi, \theta) : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ . In this setting, the partial derivative  $\nabla_{\theta}f(\phi, \theta)$  is the  $n \times m$  matrix  $(\partial_{\theta_i}f_j)_{ij}$  with  $i = 1, \dots, n$  and  $j = 1, \dots, m$ .

We aim to understand the impact of discretizing the ODEs

$$\dot{\phi} = f(\phi, \theta), \quad (2)$$

$$\dot{\theta} = g(\phi, \theta), \quad (3)$$

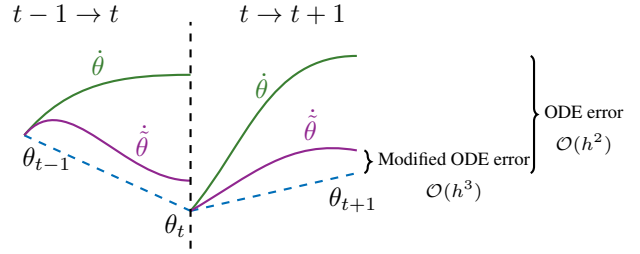


Figure 1. Visualization of our approach, given by backward error analysis. For each player (we show only  $\theta$  for simplicity), we find the modified ODE  $\dot{\theta}$  which captures the change in parameters introduced by the discrete updates with an error of  $\mathcal{O}(h^3)$ . The modified ODE  $\dot{\theta}$  follows the discrete update more closely than the original ODE  $\dot{\theta}$ , which has an error of  $\mathcal{O}(h^2)$ .

using either simultaneous or alternating Euler updates. We derive a modified continuous system of the form:

$$\dot{\phi} = f(\phi, \theta) + hf_1(\phi, \theta) \quad (4)$$

$$\dot{\theta} = g(\phi, \theta) + hg_1(\phi, \theta) \quad (5)$$

that closely follows the discrete Euler update steps; the local error between a discrete update and the modified system is of order  $\mathcal{O}(h^3)$  instead of  $\mathcal{O}(h^2)$  as is the case for the original continuous system given by Equations (2) and (3). If we can neglect errors of order  $\mathcal{O}(h^3)$ , the terms  $f_1$  and  $g_1$  above characterize the DD of the discrete scheme, which can be used to help us understand the impact of DD. For example, it allows us to compare the use of simultaneous and alternating Euler updates, by comparing the dynamics of their associated modified systems as characterized by the DD terms  $f_1$  and  $g_1$ .

We can specialize these modified equations using  $f = -\nabla_{\phi}L_1$  and  $g = -\nabla_{\theta}L_2$ , where  $L_1$  and  $L_2$  are the loss functions for the two players. We will use this setting later to investigate the modified dynamics of *simultaneous* or *alternating gradient descent*. We can further specialize the form of these updates for common-payoff games ( $L_1 = L_2 = E$ ) and zero-sum games ( $L_1 = -E, L_2 = E$ ).

### 3.1. DD for simultaneous Euler updates

The *simultaneous Euler updates* with learning rates  $\alpha h$  and  $\lambda h$  respectively are given by

$$\phi_t = \phi_{t-1} + \alpha h f(\theta_{t-1}, \phi_{t-1}) \quad (6)$$

$$\theta_t = \theta_{t-1} + \lambda h g(\theta_{t-1}, \phi_{t-1}) \quad (7)$$

**Theorem 3.1** *The discrete simultaneous Euler updates in*

(6) and (7) follow the continuous system

$$\begin{aligned}\dot{\phi} &= f - \frac{\alpha h}{2} (f \nabla_{\phi} f + g \nabla_{\theta} f) \\ \dot{\theta} &= g - \frac{\lambda h}{2} (f \nabla_{\phi} g + g \nabla_{\theta} g)\end{aligned}$$

with an error of size  $\mathcal{O}(h^3)$  after one update step.

*Remark:* A special case of Theorem 3.1 for zero-sum games with equal learning rates can be found in Lu (2021).

### 3.2. DD for alternating Euler updates

For *alternating Euler updates*, the players take turns to update their parameters, and can perform multiple updates each. We denote the number of alternating updates of the first player (resp. second player) by  $m$  (resp.  $k$ ). We scale the learning rates by the number of updates, leading to the following updates  $\phi_t := \phi_{m,t}$  and  $\theta_t := \theta_{k,t}$  where

$$\phi_{i,t} = \phi_{i-1,t} + \frac{\alpha h}{m} f(\phi_{i-1,t}, \theta_{t-1}), \quad i = 1 \dots m, \quad (8)$$

$$\theta_{j,t} = \theta_{j-1,t} + \frac{\lambda h}{k} g(\phi_{m,t}, \theta_{j-1,t}), \quad j = 1 \dots k. \quad (9)$$

**Theorem 3.2** *The discrete alternating Euler updates in (8) and (9) follow the continuous system*

$$\begin{aligned}\dot{\phi} &= f - \frac{\alpha h}{2} \left( \frac{1}{m} f \nabla_{\phi} f + g \nabla_{\theta} f \right) \\ \dot{\theta} &= g - \frac{\lambda h}{2} \left( \left(1 - \frac{2\alpha}{\lambda}\right) f \nabla_{\phi} g + \frac{1}{k} g \nabla_{\theta} g \right)\end{aligned}$$

with an error of size  $\mathcal{O}(h^3)$  after one update step.

*Remark:* Equilibria of the original continuous systems (i.e., points where  $f = \mathbf{0}$  and  $g = \mathbf{0}$ ) remain equilibria of the modified continuous systems.

**Definition 3.1** The discretization drift for each player has two terms: one term containing a player’s own update function only - terms we will call *self terms* - and a term that also contains the other player’s update function - which we will call *interaction terms*.

### 3.3. Sketch of the proofs

Following backward error analysis, the idea is to modify the original continuous system by adding corrections in powers of the learning rate:  $\tilde{f} = f + h f_1 + h^2 f_2 + \dots$  and  $\tilde{g} = g + h g_1 + h^2 g_2 + \dots$ , where for simplicity in this proof sketch, we use the same learning rate for both players (detailed derivations can be found in the Supplementary Material). We want to find corrections  $f_i, g_i$  such that the modified system  $\dot{\phi} = \tilde{f}$  and  $\dot{\theta} = \tilde{g}$  follows the discrete update steps exactly. To do that we proceed in three steps:

**Step 1:** We expand the numerical scheme to find a relationship between  $\phi_t$  and  $\phi_{t-1}$  and  $\theta_t$  and  $\theta_{t-1}$  up to order  $\mathcal{O}(h^2)$ . In the case of the simultaneous Euler updates this does not require any change to Equations (6) and (7), while for alternating updates we have to expand the intermediate steps of the integrator using Taylor series.

**Step 2:** We compute the Taylor’s series of the modified equations solution, yielding:

$$\tilde{\phi}(h) = \phi_{t-1} + h f + h^2 \left( f_1 + \frac{1}{2} (f \nabla_{\phi} f + g \nabla_{\theta} f) \right) + \mathcal{O}(h^3)$$

$$\tilde{\theta}(h) = \theta_{t-1} + h g + h^2 \left( g_1 + \frac{1}{2} (f \nabla_{\phi} g + g \nabla_{\theta} g) \right) + \mathcal{O}(h^3),$$

where all the  $f$ ’s,  $g$ ’s, and their derivatives are evaluated at  $(\phi_{t-1}, \theta_{t-1})$ .

**Step 3:** We match the terms of equal power in  $h$  so that the solution of modified equations coincides with the discrete update after one step. This amounts to finding the corrections,  $f_i$ ’s and  $g_i$ ’s, so that all the terms of order higher than  $\mathcal{O}(h)$  in the modified equation solution above will vanish; this yields the first order drift terms  $f_1$  and  $g_1$  in terms of  $f, g$ , and their derivatives. For simultaneous updates we obtain  $f_1 = -\frac{1}{2} (f \nabla_{\phi} f + g \nabla_{\theta} f)$  and  $g_1 = -\frac{1}{2} (f \nabla_{\phi} g + g \nabla_{\theta} g)$ , and by construction, we have obtained the modified truncated system which follows the discrete updates exactly up to order  $\mathcal{O}(h^3)$ , leading to Theorem 3.1.

*Remark:* The modified equations in Theorems 3.1 and 3.2 closely follow the discrete updates only for learning rates where errors of size  $\mathcal{O}(h^3)$  can be neglected. Beyond this, higher order corrections are likely to contribute to the DD.

### 3.4. Visualizing trajectories

To illustrate the effect of DD in two-player games, we use a simple example adapted from Balduzzi et al. (2018):

$$\dot{\phi} = f(\phi, \theta) = -\epsilon_1 \phi + \theta; \quad \dot{\theta} = g(\phi, \theta) = \epsilon_2 \theta - \phi \quad (10)$$

In Figure 2 we validate our theory empirically by visualizing the trajectories of the discrete Euler steps for simultaneous and alternating updates, and show that they closely match the trajectories of the corresponding modified continuous systems that we have derived. To visualize the trajectories of the original unmodified continuous system, we use Runge-Kutta 4 (a fourth-order numerical integrator that has no DD up to  $\mathcal{O}(h^5)$  in the case where the same learning rates are used for the two players — see Hairer & Lubich (1997) and Supplementary Material for proofs). We will use Runge-Kutta 4 as a baseline throughout the paper.

## 4. The stability of DD

The long-term behavior of gradient-based training can be characterized by the stability of its equilibria. Using stability

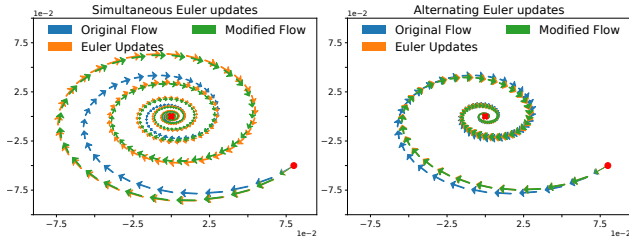


Figure 2. The modified continuous flow captures the effect of DD in simultaneous and alternating Euler updates.

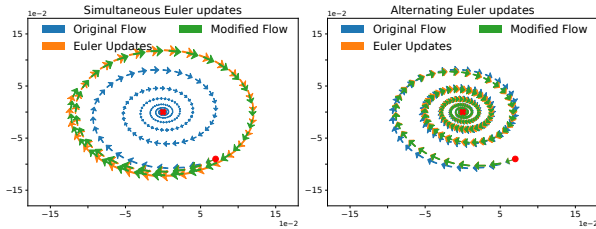


Figure 3. Discretization drift can change the stability of a game.  $\epsilon_1 = \epsilon_2 = 0.09$  and a learning rate 0.2.

analysis of a continuous system to understand discrete dynamics in two-player games has a fruitful history; however, prior work ignored discretization drift, since they analyse the stability of the original game ODEs. Using the modified ODEs given by backward error analysis provides two benefits here: 1) they account for the discretization drift, and 2) they provide *different* ODEs for simultaneous and for alternating updates, capturing the specificities of the two optimizers.

The modified ODE approach gives us a method to analyze the stability of the discrete updates: 1) Choose the system and the update type – simultaneous or alternating – to be analyzed; 2) write the modified ODEs for the chosen system as given by Theorems 3.1 and 3.2; 3) write the corresponding modified Jacobian, and evaluate it at an equilibrium; 4) determine the stability of the equilibrium by computing the eigenvalues of the modified Jacobian.

Steps 1 and 2 are easy, and step 4 is required in the stability analysis of any continuous system. For step 3, we provide a general form of the modified Jacobian at the equilibrium point of the original system, where  $f = \mathbf{0}$  and  $g = \mathbf{0}$ :

$$\tilde{J} = \begin{bmatrix} \nabla_{\phi} \tilde{f} & \nabla_{\theta} \tilde{f} \\ \nabla_{\phi} \tilde{g} & \nabla_{\theta} \tilde{g} \end{bmatrix} = J - \frac{h}{2} K \quad (11)$$

where  $J$  is the unmodified Jacobian and  $K$  is a matrix that depends on the update type. For simultaneous updates (see Supplementary Material for alternating updates)  $K =$

$$\begin{bmatrix} \alpha(\nabla_{\phi} f)^2 + \alpha \nabla_{\phi} g \nabla_{\theta} f & \alpha \nabla_{\theta} f \nabla_{\phi} f + \alpha \nabla_{\theta} g \nabla_{\theta} f \\ \lambda \nabla_{\phi} g \nabla_{\theta} g + \lambda \nabla_{\phi} f \nabla_{\theta} g & \lambda(\nabla_{\theta} g)^2 + \lambda \nabla_{\theta} f \nabla_{\theta} g \end{bmatrix}.$$

Using the method above we show (in the Supplementary Material), that, in two-player games, the drift can change a stable equilibrium into an unstable one, which is not the case for supervised learning (Barrett & Dherin, 2021). For simultaneous updates with equal learning rates, this recovers a result of Daskalakis & Panageas (2018) derived in the context of zero-sum games. We show this by example: consider the game given by the system of ODEs in Equation (10) with  $\epsilon_1 = \epsilon_2 = 0.09$ . The stability analysis of its modified ODEs for the simultaneous Euler updates shows they diverge when  $\alpha h = \lambda h = 0.2$ . We use the same example to illustrate the difference in behavior between simultaneous and alternating updates: the stability analysis shows the modified ODEs for alternating updates converge to a stable equilibrium. In both cases, the results obtained using the stability analysis of the modified ODEs is consistent with empirical outcomes obtained by following the corresponding discrete updates, as shown in Figure 3; this would not have been the case had we used the original system to do stability analysis, which would have always predicted convergence to an equilibrium.

The modified ODEs help us bridge the gap between theory and practice: they allow us to extend the reach of stability analysis to a wider range of techniques used for training, such as alternating gradient descent. We hope the method we provide will be used in the context of GANs, to expand prior work such as that of Nagarajan & Kolter (2017) to alternating updates. However, the modified ODEs are not without limitations: they ignore discretization errors smaller than  $\mathcal{O}(h^3)$ , and thus they are not equivalent to the discrete updates; methods that directly assess the convergence of discrete updates (e.g. Mescheder et al. (2017)) remain an indispensable tool for understanding discrete systems.

## 5. Common-payoff games

When the players share a common loss, as in *common-payoff games*, we recover supervised learning with a single loss  $E$ , but with the extra-freedom of training the weights corresponding to different parts of the model with possibly different learning rates and update strategies (see for instance You et al. (2018) where a per-layer learning rate is used to obtain extreme training speedups at equal levels of test accuracy). A special case occurs when the two players with equal learning rates ( $\alpha = \lambda$ ) perform simultaneous gradient descent. In this case, both modified losses exactly recover Equation (1). Barrett & Dherin (2021) argue that minimizing the loss-gradient norm, in this case, has a beneficial effect.

In this section, we instead focus on alternating gradient descent. We partition a neural network into two parts, corresponding to two sets of parameters,  $\phi$  for the parameters closer to the input and  $\theta$  for the parameters closer to the output. This procedure freezes one part of the network while

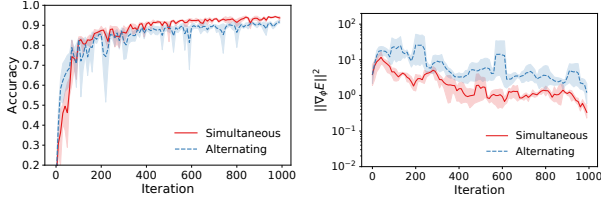


Figure 4. In common-payoff games alternating updates lead to higher gradient norms and unstable training.

training the other part and alternating between the two parts. This scenario may arise in a distributed training setting, as a form of block coordinate descent. In the case of common-payoff games, we have the following as a special case of Theorem 3.2 by substituting  $f = -\nabla_{\phi} E$  and  $g = -\nabla_{\theta} E$ :

**Corollary 5.1** *In a two-player common-payoff game with common loss  $E$ , alternating gradient descent – as described in equations (8) and (9) – with one update per player follows a gradient flow given by the modified losses*

$$\tilde{L}_1 = E + \frac{\alpha h}{4} \left( \|\nabla_{\phi} E\|^2 + \|\nabla_{\theta} E\|^2 \right) \quad (12)$$

$$\tilde{L}_2 = E + \frac{\lambda h}{4} \left( \left(1 - \frac{2\alpha}{\lambda}\right) \|\nabla_{\phi} E\|^2 + \|\nabla_{\theta} E\|^2 \right) \quad (13)$$

with an error of size  $\mathcal{O}(h^3)$  after one update step.

The term  $\left(1 - \frac{2\alpha}{\lambda}\right) \|\nabla_{\phi} E\|^2$  in Eq. (13) is negative when the learning rates are equal, seeking to maximize the gradient norm of player  $\phi$ . According to Barrett & Dherin (2021), we expect less stable training and worse performance in this case. This prediction is verified in Figure 4, where we compare simultaneous and alternating gradient descent for a MLP trained on MNIST with a common learning rate.

## 6. Analysis of zero-sum games

We now study zero-sum games, where the two adversarial players have opposite losses  $-L_1 = L_2 = E$ . We substitute the updates  $f = \nabla_{\phi} E$  and  $g = -\nabla_{\theta} E$  in the Theorems in Section 3 and obtain:

**Corollary 6.1** *In a zero-sum two-player differentiable game, simultaneous gradient descent updates – as described in equations (6) and (7) – follows a gradient flow given by the modified losses*

$$\tilde{L}_1 = -E + \frac{\alpha h}{4} \|\nabla_{\phi} E\|^2 - \frac{\alpha h}{4} \|\nabla_{\theta} E\|^2, \quad (14)$$

$$\tilde{L}_2 = E - \frac{\lambda h}{4} \|\nabla_{\phi} E\|^2 + \frac{\lambda h}{4} \|\nabla_{\theta} E\|^2, \quad (15)$$

with an error of size  $\mathcal{O}(h^3)$  after one update step.

*Remark:* Discretization drift preserves the adversarial structure of the game, with the *interaction terms* maximizing the

gradient norm of the opposite player, while the *self terms* are minimizing the player’s own gradient norm.

*Remark:* The modified losses of zero-sum games trained with simultaneous gradient descent with different learning rates are no longer zero-sum.

**Corollary 6.2** *In a zero-sum two-player differentiable game, alternating gradient descent – as described in equations (8) and (9) – follows a gradient flow given by the modified losses*

$$\tilde{L}_1 = -E + \frac{\alpha h}{4m} \|\nabla_{\phi} E\|^2 - \frac{\alpha h}{4} \|\nabla_{\theta} E\|^2 \quad (16)$$

$$\tilde{L}_2 = E - \frac{\lambda h}{4} \left(1 - \frac{2\alpha}{\lambda}\right) \|\nabla_{\phi} E\|^2 + \frac{\lambda h}{4k} \|\nabla_{\theta} E\|^2 \quad (17)$$

with an error of size  $\mathcal{O}(h^3)$  after one update step.

*Remark:* The modified losses of zero-sum games trained with alternating gradient descent are not zero sum.

*Remark:* Since  $\left(1 - \frac{2\alpha}{\lambda}\right) < 1$  in the alternating case there is always less weight on the term encouraging maximizing  $\|\nabla_{\phi} E\|^2$  compared to the simultaneous case under the same learning rates. For  $\frac{\alpha}{\lambda} > \frac{1}{2}$  both players minimize  $\|\nabla_{\phi} E\|^2$ .

### 6.1. Dirac-GAN: an illustrative example

Mescheder et al. (2018) introduce the Dirac-GAN as an example to illustrate the often complex training dynamics in zero-sum games. We follow this example to provide an intuitive understanding of DD. The generative adversarial network (Goodfellow et al., 2014) is an example of two-player game that has been successfully used for distribution learning. The generator  $G$  with parameters  $\theta$  learns a mapping from samples of the latent distribution  $\mathbf{z} \sim p(\mathbf{z})$  to the data space, while the discriminator  $D$  with parameters  $\phi$  learns to distinguish these samples from data. Dirac-GAN aims to learn a Dirac delta distribution with mass at zero; the generator is modeling a Dirac with parameter  $\theta$ :  $G(z) = \theta$  and the discriminator is a linear model on the input with parameter  $\phi$ :  $D_{\phi}(x) = \phi x$ . This results in the zero-sum game given by:

$$E = l(\theta\phi) + l(0) \quad (18)$$

where  $l$  depends on the GAN formulation used ( $l(z) = -\log(1 + e^{-z})$  for instance). The unique equilibrium point is  $\theta = \phi = 0$ . All proofs for this section are in the Supplementary Material, together with visualizations.

*Reconciling discrete and continuous updates in Dirac-GAN:* The continuous dynamics induced by the gradient field from Equation (18) preserve  $\theta^2 + \phi^2$ , while for simultaneous gradient descent  $\theta^2 + \phi^2$  increases with each update (Mescheder et al., 2018); different conclusions are reached when analyzing the dynamics of the original continuous system versus

the discrete updates. We show that the modified ODEs given by equations (14) and (15) resolve this discrepancy, since simultaneous gradient descent *modifies* the continuous dynamics to increase  $\theta^2 + \phi^2$ , leading to consistent conclusions from the modified continuous and discrete dynamics.

*Explicit regularization stabilizes Dirac-GAN:* To counteract the instability in the Dirac-GAN induced by the interaction terms we can add explicit regularization with the same functional form:  $L_1 = -E + u\|\nabla_{\theta}E\|^2$  and  $L_2 = E + \nu\|\nabla_{\phi}E\|^2$  where  $u, \nu$  are of  $\mathcal{O}(h)$ . We find that the modified Jacobian for this modified system with explicit regularization is negative definite if  $u > h\alpha/4$  and  $\nu > h\lambda/4$ , so the system is asymptotically stable and converges to the optimum. Notably, by quantifying DD, we are able to find the regularization coefficients which guarantee convergence and show that they depend on learning rates.

## 7. Experimental analysis of GANs

To understand the effect of DD on more complex adversarial games, we analyze GANs trained for image generation on the CIFAR10 dataset. We follow the model architectures from Spectral-Normalized GANs (Miyato et al., 2018). Both players have millions of parameters. We employ the original GAN formulation, where the discriminator is a binary classifier trained to classify data from the dataset as real and model samples as fake; the generator tries to generate samples that the discriminator considers as real. This can be formulated as a zero-sum game:

$$E = \mathbb{E}_{p^*(\mathbf{x})} \log D_{\phi}(\mathbf{x}) + \mathbb{E}_{p_{\theta}(\mathbf{z})} \log(1 - D_{\phi}(G_{\theta}(\mathbf{z})))$$

When it comes to gradient descent, GAN practitioners often use alternating, not simultaneous updates: the discriminator is updated first, followed by the generator. However, recent work shows higher-order numerical integrators can work well with simultaneous updates (Qin et al., 2020). We will show that DD can be seen as the culprit behind some of the challenges in simultaneous gradient descent in zero-sum GANs, indicating ways to improve training performance in this setting. For a clear presentation of the effects of DD, we employ a minimalist training setup. Instead of using popular adaptive optimizers such as Adam (Kingma & Ba, 2015), we train all the models with vanilla stochastic gradient descent, without momentum or variance reduction methods.

We use the Inception Score (IS) (Salimans et al., 2016) for evaluation. Our training curves contain a horizontal line at the Inception Score of 7.5, obtained with the same architectures we use, but with the Adam optimizer (the score reported by Miyato et al. (2018) is 7.42). *All learning curves correspond to the best 10% of models* for the corresponding training setting from a sweep over learning rates and 5 seeds - for a discussion of variance across seeds

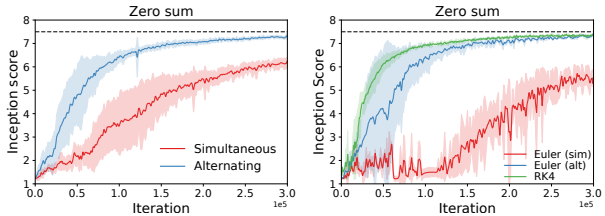


Figure 5. The effect of DD on zero-sum games. Alternating updates perform better (left), and with equal learning rates (right), RK4 has  $\mathcal{O}(h^5)$  drift and performs better.

and corresponding plots, as well as best top 20% and 30% of models see the Supplementary Material. We also report box plots showing the performance quantiles across all hyperparameter and seeds, together with the top 10% of models. For SGD we use learning rates  $\{5 \times 10^{-2}, 1 \times 10^{-2}, 5 \times 10^{-3}, 1 \times 10^{-3}\}$  for each player; for Adam, we use learning rates  $\{1 \times 10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}, 4 \times 10^{-4}\}$ , which have been widely used in the literature (Miyato et al., 2018). When comparing to Runge-Kutta (RK4) to assess the effect of DD we always use the same learning rates for both players. We present results using additional losses, via LS-GAN (Mao et al., 2017), and report FID results (Heusel et al., 2017) and full experimental details in the Supplementary Material. The code associated with this work can be found at [https://github.com/deepmind/deepmind-research/dd\\_two\\_player\\_games](https://github.com/deepmind/deepmind-research/dd_two_player_games).

### 7.1. Does DD affect training?

We start our experimental analysis by showing the effect of DD on zero-sum games. We compare simultaneous gradient descent, alternating gradient descent and Runge-Kutta 4 updates, since they follow different continuous dynamics given by the modified equations we have derived up to  $\mathcal{O}(h^3)$  error. Figure 5 shows simultaneous gradient descent performs substantially worse than alternating updates. When compared to Runge-Kutta 4, which has a DD error of  $\mathcal{O}(h^5)$  when the two players have equal learning rates, we see that Runge-Kutta performs better, and that removing the drift improves training. Multiple updates also affect training, either positively or negatively, depending on learning rates - see Supplementary Material.

### 7.2. The importance of learning rates in DD

While in simultaneous updates (Eqs (14) and (15)) the interaction terms of both players maximize the gradient norm of the other player, alternating gradient descent (Eqs (16) and (17)) exhibits less pressure on the second player (generator) to maximize the norm of the first player (discriminator). In alternating updates, when the ratio between the discriminator and generator learning rates exceeds 0.5, both play-

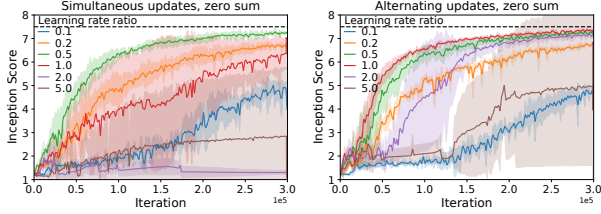


Figure 6. Alternating gradient descent performs better for learning rate ratios which reduce the adversarial nature of DD. The same learning rate ratios show no advantage in the simultaneous case.

ers are encouraged to minimize the discriminator’s gradient norm. To understand the effect of learning rate ratios in training, we performed a sweep where the discriminator learning rates  $\alpha$  are sampled uniformly between  $[0.001, 0.01]$ , and the learning rate ratios  $\alpha/\lambda$  are in  $\{0.1, 0.2, 0.5, 1, 2, 5\}$ , with results shown in Figure 6. Learning rate ratios greater than 0.5 perform best for alternating updates, and enjoy a substantial increase in performance compared to simultaneous updates.

### 7.3. Improving performance by explicit regularization

We investigate whether *canceling the interaction terms* between the two players can improve training stability and performance in zero-sum games trained using simultaneous gradient descent. We train models using the losses:

$$L_1 = -E + c_1 \|\nabla_{\theta} E\|^2 \quad (19)$$

$$L_2 = E + c_2 \|\nabla_{\phi} E\|^2 \quad (20)$$

If  $c_1, c_2$  are  $\mathcal{O}(h)$  we can ignore the DD from these regularization terms, since their effect on DD will be of order  $\mathcal{O}(h^3)$ . We can set coefficients to be the negative of the coefficients present in DD, namely  $c_1 = \alpha h/4$  and  $c_2 = \lambda h/4$ ; we thus cancel the interaction terms which maximize the gradient norm of the other player, while keeping the self terms, which minimize the player’s own gradient norm. We show results in Figure 7: canceling the interaction terms leads to substantial performance improvement compared to SGD, obtains the same peak performance as Adam (though requires more training iterations) and recovers the performance of Runge-Kutta 4 (Figure 5). Unlike Adam, we do not observe a decrease in performance when trained for longer but report higher variance in performance across seeds - see Supplementary Material.

*Connection with Symplectic Gradient Adjustment (SGA):* Balduzzi et al. (2018) proposed SGA to improve the dynamics of gradient-based method for games, by counter-acting the rotational force of the vector field. Adjusting the gradient field can be viewed as modifying the losses as in Equation (20); the modification from SGA cancels the interaction terms we identified. However, it is unclear whether

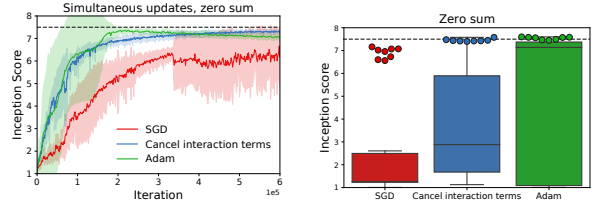
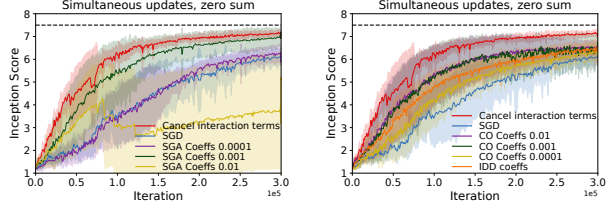


Figure 7. Simultaneous updates: Explicit regularization canceling the interaction terms of DD improves performance, both for the best performing models (left) and across a sweep (right).



(a) SGA comparison.

(b) CO comparison.

Figure 8. Comparison with Symplectic Gradient Adjustment (SGA) and Consensus Optimization (CO): DD motivates the form of explicit regularization and provides the optimal regularization coefficients, without requiring a larger hyperparameter sweep.

the fixed coefficients of  $c_1 = c_2 = \frac{1}{2}$  used by SGA are always optimal while our analysis shows the strength of DD changes with the exact discretization scheme, such as the step size. Indeed, as our experimental results in Figure 8(a) show, adjusting the coefficient in SGA strongly affects training.

*Connection with Consensus Optimization (CO):* Mescheder et al. (2017) analyze the discrete dynamics of gradient descent in zero-sum games and prove that, under certain assumptions, adding explicit regularization that encourages the players to minimize the gradient norm of both players guarantees convergence to a local Nash equilibrium. Their approach includes canceling the interaction terms, but also requires *strengthening the self terms*, using losses:

$$L_1 = -E + c_1 \|\nabla_{\theta} E\|^2 + s_1 \|\nabla_{\phi} E\|^2 \quad (21)$$

$$L_2 = E + s_2 \|\nabla_{\theta} E\|^2 + c_2 \|\nabla_{\phi} E\|^2, \quad (22)$$

where they use  $s_1 = s_2 = c_1 = c_2 = \gamma$  where  $\gamma$  is a hyperparameter. In order to understand the effect of the self and interaction terms, we compare to CO, as well as a similar approach where we use coefficients proportional to the drift, namely  $s_1 = \alpha h/4$  and  $s_2 = \lambda h/4$ ; this effectively doubles the strength of the self terms in DD. We show results in Figure 8(b). We first notice that CO can improve results over vanilla SGD. However, similarly to what we observed with SGA, the regularization coefficient is important and thus requires a hyperparameter sweep, unlike our approach which

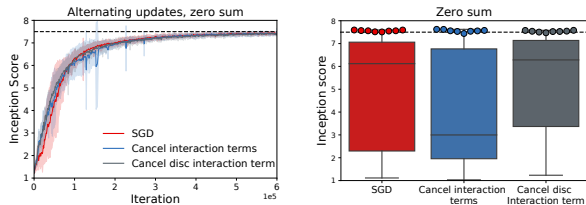


Figure 9. Alternating updates: DD tells us which explicit regularization terms to use; canceling only the discriminator interaction term improves sensitivity across hyperparameters (right).

uses the coefficients provided by the DD. We further notice that strengthening the norms using the DD coefficients can improve training, but performs worse compared to only canceling the interaction terms. This shows the importance of finding the right training regime, and that strengthening the self terms does not always improve performance.

*Alternating updates:* We perform the same exercise for alternating updates, where  $c_1 = \alpha h/4$  and  $c_2 = \lambda h/4(1 - \frac{2\alpha}{\lambda})$ . We also study the performance obtained by only canceling the discriminator interaction term, since when  $\alpha/\lambda > 0.5$  the generator interaction term minimizes, rather than maximizes, the discriminator gradient norm and thus the generator interaction term might not have a strong destabilizing force. We observe that adding explicit regularization mainly brings the benefit of reduced variance when canceling the discriminator interaction term (Figure 9). As for simultaneous updates, we find that knowing the form of DD guides us to a choice of explicit regularization: for alternating updates canceling both interaction terms can hurt training, but the form of the modified losses suggests that we should only cancel the discriminator interaction term, with which we can obtain some gains.

*Does canceling the interaction terms help for every choice of learning rates?* The substantial performance and stability improvement we observe applies to the performance obtained across a learning rate sweep. For individual learning rate choices however, canceling the interaction terms is not guaranteed to improve learning.

#### 7.4. Extension to non-zero-sum GANs

Finally, we extend our analysis to GANs with the non-saturating loss for the generator  $E_G = -\log D_\phi(G_\theta(\mathbf{z}))$  introduced by Goodfellow et al. (2014), while keeping the discriminator loss unchanged as  $E_D = \mathbb{E}_{p^*(\mathbf{x})} \log D_\phi(\mathbf{x}) + \mathbb{E}_{p_\theta(\mathbf{z})} \log(1 - D_\phi(G_\theta(\mathbf{z})))$ . In contrast with the dynamics from zero-sum GANs we analyzed earlier, changing from simultaneous to alternating updates results in little change in performance - as can be seen in Figure 10. Despite having the same adversarial structure and the same discriminator loss, changing the generator loss changes the relative perfor-

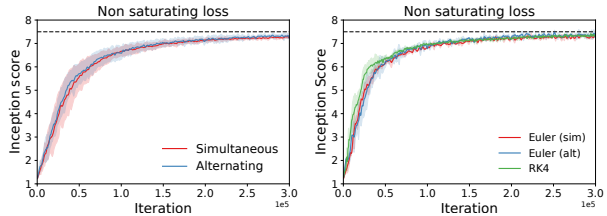


Figure 10. The effect of DD depends on the game: its effect is less strong for the non-saturating loss.

mance of the different discrete update schemes. Since the effect of DD strongly depends on the game, we recommend analyzing the performance of discrete numerical schemes on a case by case basis. Indeed, while for general two-player games we cannot always write modified losses as for the zero-sum case – see the Supplementary Material for a discussion – we can use Theorems 3.1 and 3.2 to understand the effect of the drift for specific choices of loss functions.

## 8. Related work

**Backward error analysis:** There have been a number of recent works applying backward error analysis to machine learning in the one-player case (supervised learning), which can potentially be extended to the two-players settings. In particular, Smith et al. (2021) extended the analysis of Barrett & Dherin (2021) to stochastic gradient descent. Kunin et al. (2021) used modified gradient flow equations to show that discrete updates break certain conservation laws present in the gradient flows of deep learning models. Franca et al. (2020) compare momentum and Nesterov acceleration using their modified equations. França et al. (2021) used backward error analysis to help devise optimizers with a control on their stability and convergence rates. Li et al. (2017) used modified equation techniques in the context of stochastic differential equations to devise optimizers with adaptive learning rates. Other recent works such as Unlu & Aitchison (2020) and Jastrzebski et al. (2020) noticed the strong impact of implicit gradient regularization in the training of over-parametrized models with SGD using different approaches. These works provide evidence that DD has a significant impact in deep learning, and that backward error analysis is a powerful tool to quantify it. At last, note that a special case of Theorem 3.1 for zero-sum games with equal learning rates can be found in Lu (2021).

**Two-player games:** As one of the best-known examples at the interface of game theory and deep learning, GANs have been powered by gradient-based optimizers as other deep neural networks. The idea of an implicit regularization effect induced by simultaneous gradient descent in GAN training was first discussed in Schäfer et al. (2020); the authors show empirically that the implicit regulariza-



	Explicit	First player ( $\phi$ )	Second player ( $\theta$ )
DD (S)	$\times$	$\frac{\alpha h}{4} \ \nabla_{\phi} E\ ^2 - \frac{\alpha h}{4} \ \nabla_{\theta} E\ ^2$	$-\frac{\lambda h}{4} \ \nabla_{\phi} E\ ^2 + \frac{\lambda h}{4} \ \nabla_{\theta} E\ ^2$
DD (A)	$\times$	$\frac{\alpha h}{4m} \ \nabla_{\phi} E\ ^2 - \frac{\alpha h}{4} \ \nabla_{\theta} E\ ^2$	$\frac{(2\alpha-\lambda)h}{4} \ \nabla_{\phi} E\ ^2 + \frac{\lambda h}{4k} \ \nabla_{\theta} E\ ^2$
Cancel DD interaction terms (S)	$\checkmark$	$\frac{\alpha h}{4} \ \nabla_{\theta} E\ ^2$	$\frac{\lambda h}{4} \ \nabla_{\phi} E\ ^2$
Cancel DD interaction terms (A)	$\checkmark$	$\frac{\alpha h}{4} \ \nabla_{\theta} E\ ^2$	$-\frac{(2\alpha-\lambda)h}{4} \ \nabla_{\phi} E\ ^2$
SGA (S)	$\checkmark$	$\frac{1}{2} \ \nabla_{\theta} E\ ^2$	$\frac{1}{2} \ \nabla_{\phi} E\ ^2$
Consensus optimization (S)	$\checkmark$	$\eta \ \nabla_{\phi} E\ ^2 + \eta \ \nabla_{\theta} E\ ^2$	$\eta \ \nabla_{\phi} E\ ^2 + \eta \ \nabla_{\theta} E\ ^2$
Locally stable GAN (S)	$\checkmark$	$\times$	$\eta \ \nabla_{\phi} E\ ^2$
ODE-GAN (S)	$\checkmark$	$\eta \ \nabla_{\theta} E\ ^2$	$\times$

Table 1. Comparing DD with explicit regularization methods in zero-sum games. SGA (without alignment, see the Supplementary Material and Balduzzi et al. 2018), Consensus Optimization (Mescheder et al., 2017), Locally Stable GAN (Nagarajan & Kolter, 2017), ODE-GAN (Qin et al., 2020). We assume a  $\min_{\theta} \min_{\phi}$  game with learning rates  $\alpha h$  and  $\lambda h$  and number of updates  $m$  and  $k$ , respectively. S and A denote simultaneous and alternating updates.

tion induced by gradient descent can have a positive effect on GAN training, and take a game theoretic approach to strengthening it using competitive gradient descent (Schäfer & Anandkumar, 2019). By quantifying the DD induced by gradient descent using backward error analysis, we shed additional light on the regularization effect that discrete updates have on GAN training, and show it has both beneficial and detrimental components (as also shown in Daskalakis & Panageas (2018) with different methods in the case of simultaneous GD). Moreover, we show that the explicit regularization inspired by DD results in drastically improved performance. The form of the modified losses we have derived are related to explicit regularization for GANs, which has been one of the most efficient methods for stabilizing GANs as well as other games. Some of the regularizers constrain the complexity of the players (Miyato et al., 2018; Gulrajani et al., 2017; Brock et al., 2018), while others modify the dynamics for better convergence properties (Qin et al., 2020; Mescheder et al., 2018; Nagarajan & Kolter, 2017; Balduzzi et al., 2018; Wang et al., 2019; Mazumdar et al., 2019). Our approach is orthogonal in that, with backward analysis, we start from understanding the most basic gradient steps, underpinning any further modifications of the losses or gradient fields. Importantly, we discovered a relationship between learning rates and the underlying regularization. Since merely canceling the effect of DD is insufficient in practice (as also observed by Qin et al. (2020)), our approach complements regularizers that are explicitly designed to improve convergence. We leave to future research to further study other regularizers in combination with our analysis. We summarize some of these regularizers in Table 1.

To our knowledge, this is the first work towards finding continuous systems which better match the gradient descent updates used in two player games. Studying discrete algorithms using continuous systems has a rich history in optimization (Su et al., 2016; Wibisono et al., 2016). Re-

cently, models that directly parametrize differential equations demonstrated additional potential from bridging the discrete and continuous perspectives (Chen et al., 2018; Grathwohl et al., 2018).

## 9. Discussion

We have shown that using modified continuous systems to quantify the discretization drift induced by gradient descent updates can help bridge the gap between the discrete optimization dynamics used in practice and the analysis of the original systems through continuous methods. This allowed us to cast a new light on the stability and performance of games trained using gradient descent, and guided us towards explicit regularization strategies inspired by these modified systems. We note however that DD merely modifies a game’s original dynamics, and that the DD terms alone can not fully characterize a discrete scheme. In this sense, our approach only complements works analyzing the underlying game (Shoham & Leyton-Brown, 2008). Also, it is worth noting that our analysis is valid for learning rates small enough that errors of size  $\mathcal{O}(h^3)$  can be neglected.

We have focused our empirical analysis on a few classes of two-player games, but the effect of DD will be relevant for all games trained using gradient descent. Our method can be expanded beyond gradient descent to other optimization methods such as Adam (Kingma & Ba, 2015), as well as to the stochastic setting as shown in Smith et al. (2021). We hope that our analysis of gradient descent provides a useful building block to further the understanding and performance of two-player games.

**Acknowledgments.** We would like to thank the reviewers and area chair for useful suggestions, and Alex Goldin, Chongli Qin, Jeff Donahue, Marc Deisenroth, Michael Munn, Patrick Cole, Sam Smith and Shakir Mohamed for useful discussions and support throughout this work.

---

## References

- Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., and Graepel, T. The mechanics of n-player differentiable games. In *International Conference on Machine Learning*, pp. 354–363. PMLR, 2018.
- Barrett, D. G. and Dherin, B. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021.
- Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.
- Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In *Advances in neural information processing systems*, pp. 6571–6583, 2018.
- Daskalakis, C. and Panageas, I. The limit points of (optimistic) gradient descent in min-max optimization. In *Advances in Neural Information Processing Systems*, volume 31, 2018.
- Franca, G., Sulam, J., Robinson, D., and Vidal, R. Conformal symplectic and relativistic optimization. In *Conference on Neural Information Processing Systems (NeurIPS 2020)*. 2020.
- França, G., Jordan, M. I., and Vidal, R. On dissipative symplectic integration with applications to gradient-based optimization. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(4):043402, 2021.
- 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.
- Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.
- 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.
- Hairer, E. and Lubich, C. The life-span of backward error analysis for numerical integrators. *Numerische Mathematik*, 76:441–457, 06 1997.
- Hairer, E., Lubich, C., and Wanner, G. *Geometric numerical integration*, volume 31. Springer-Verlag, Berlin, second edition, 2006. ISBN 3-540-30663-3; 978-3-540-30663-4.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.
- Jastrzebski, S. K., Arpit, D., Åstrand, O., Kerg, G., Wang, H., Xiong, C., Socher, R., Cho, K., and Geras, K. J. Catastrophic fisher explosion: Early phase fisher matrix impacts generalization. *arXiv preprint arXiv:2012.14193*, 2020.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. 2015.
- Kunin, D., Sagastuy-Brena, J., and Ganguli, Surya Daniel L.K. Yamins, H. T. Symmetry, conservation laws, and learning dynamics in neural networks. In *International Conference on Learning Representations*, 2021.
- Li, Q., Tai, C., and E, W. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, volume 70, pp. 2101–2110, 2017.
- Lu, H. An  $o(sr)$ -resolution ode framework for understanding discrete-time algorithms and applications to the linear convergence of minimax problems. *arXiv preprint arXiv:2001.08826*, 2021.
- 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.
- Mazumdar, E. V., Jordan, M. I., and Sastry, S. S. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.
- 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? In *International conference on machine learning*, pp. 3481–3490. PMLR, 2018.
- Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 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.

- 
- Qin, C., Wu, Y., Springenberg, J. T., Brock, A., Donahue, J., Lillicrap, T. P., and Kohli, P. Training generative adversarial networks by solving ordinary differential equations. 2020.
- Rajeswaran, A., Mordatch, I., and Kumar, V. A game theoretic framework for model based reinforcement learning. In *International Conference on Machine Learning*, pp. 7953–7963. PMLR, 2020.
- 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.
- Schäfer, F. and Anandkumar, A. Competitive gradient descent. In *Advances in Neural Information Processing Systems*, pp. 7625–7635, 2019.
- Schäfer, F., Zheng, H., and Anandkumar, A. Implicit competitive regularization in gans. In *International Conference on Machine Learning*, pp. 8533–8544. PMLR, 2020.
- Shoham, Y. and Leyton-Brown, K. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.
- Singh, S. P., Kearns, M. J., and Mansour, Y. Nash convergence of gradient dynamics in general-sum games. In *UAI*, pp. 541–548, 2000.
- Smith, S. L., Dherin, B., Barrett, D. G., and De, S. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021.
- Su, W., Boyd, S., and Candes, E. J. A differential equation for modeling nesterov’s accelerated gradient method: theory and insights. *The Journal of Machine Learning Research*, 17(1):5312–5354, 2016.
- Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.
- Unlu, A. and Aitchison, L. Gradient regularisation as approximate variational inference. In *Symposium on Advances in Approximate Bayesian Inference*, 2020.
- Wang, Y., Zhang, G., and Ba, J. On solving minimax optimization locally: A follow-the-ridge approach. In *International Conference on Learning Representations*, 2019.
- Wibisono, A., Wilson, A. C., and Jordan, M. I. A variational perspective on accelerated methods in optimization. *proceedings of the National Academy of Sciences*, 113(47): E7351–E7358, 2016.
- You, Y., Zhang, Z., Hsieh, C.-J., Demmel, J., and Keutzer, K. Imagenet training in minutes. In *International Conference on Parallel Processing*, pp. 1–10, 2018.