The Mechanics of n-Player Differentiable Games

David Balduzzi [1] **Sébastien Racanière** [1] **James Martens** [1] **Jakob Foerster** [2] **Karl Tuyls** [1] **Thore Graepel** [1]

Abstract

The cornerstone underpinning deep learning is the guarantee that gradient descent on an objective converges to local minima. Unfortunately, this guarantee fails in settings, such as generative adversarial nets, where there are multiple interacting losses. The behavior of gradient-based methods in games is not well understood – and is becoming increasingly important as adversarial and multi-objective architectures proliferate. In this paper, we develop new techniques to understand and control the dynamics in general games. The key result is to decompose the second-order dynamics into two components. The first is related to potential games, which reduce to gradient descent on an implicit function; the second relates to *Hamiltonian games*, a new class of games that obey a conservation law, akin to conservation laws in classical mechanical systems. The decomposition motivates *Symplectic Gradient Adjustment* (SGA), a new algorithm for finding stable fixed points in general games. Basic experiments show SGA is competitive with recently proposed algorithms for finding stable fixed points in GANs – whilst at the same time being applicable to – and having guarantees in – much more general games.

1. Introduction

Recent progress in machine learning is heavily dependent on using gradient descent, applied to optimize the parameters of models with respect to a (single) objectives. A basic result is that gradient descent converges to a local minimum of the objective under a broad range of conditions (Lee et al., 2017). However, there is a rapidly growing set of powerful models that do not optimize a single objective, including: generative adversarial networks (Goodfellow et al., 2014), proximal gradient TD learning (Liu et al., 2016), multi-level

[1]DeepMind [2]University of Oxford. Correspondence to: David Balduzzi <dbalduzzi@google.com>.

optimization (Pfau & Vinyals, 2016), synthetic gradients (Jaderberg et al., 2017), hierarchical reinforcement learning (Wayne & Abbott, 2014; Vezhnevets et al., 2017), curiosity (Pathak et al., 2017), and imaginative agents (Racanière et al., 2017). In effect, the models are trained via games played by cooperating and competing modules.

No-regret algorithms such as gradient descent are guaranteed to converge to coarse correlated equilibria in games (Stoltz & Lugosi, 2007). However, the dynamics do not converge to Nash equilibria – and do not even stabilize – in general (Mertikopoulos et al., 2018). Concretely, cyclic behaviors emerge even in simple cases, see example 1.

This paper presents an analysis of the second-order structure of game dynamics that allows to identify two classes of games, potential and Hamiltonian, that are easy to solve separately. We then derive *symplectic gradient adjustment* (SGA), a method for finding stable fixed points in games. SGA's performance is evaluated in basic experiments.

Background and problem description. Procedures that converge to Nash equilibria have been found for restricted game classes: potential games, 2-player zero-sum games and a few others (Hu & Wellman, 2003; Hart & Mas-Colell, 2013). Finding equilibria can be reformulated as a nonlinear complementarity problem, but these are 'hopelessly impractical to solve' in general (Shoham & Leyton-Brown, 2008) because the problem is PPAD hard (Daskalakis et al., 2009).

Players are primarily neural nets in our setting. We therefore restrict to gradient-based methods (game-theorists have considered a much broader range of techniques). Losses are not necessarily convex in *any* of their parameters, so Nash equilibria do not necessarily exist. Leaving existence aside, finding Nash equilibria is analogous to, but much harder than, finding global minima in neural nets – which is not realistic with gradient-based methods.

There are (at least) three problems with gradient descent in games. Firstly, the potential existence of cycles (recurrent dynamics) implies there are no convergence guarantees, see example 1 and Mertikopoulos et al. (2018). Secondly, even when gradient descent converges, the rate may be too slow in practice because 'rotational forces' necessitate extremely small learning rates (see figure 3). Finally, since there is no single objective, there is no way to measure

progress. Application-specific proxies have been proposed, for example the inception score for GANs (Salimans et al., 2016), but these are little help during training – the inception score is no substitute for looking at samples.

Outline and summary of main contributions. We start with the well-known case of a zero-sum bimatrix game: example 1. It turns out that the dynamics (that is, the dynamics under simultaneous gradient descent) can be reformulated via Hamilton's equations. The cyclic behavior arises because the dynamics live on the level sets of the Hamiltonian. More directly useful, gradient descent on the Hamiltonian converges to a Nash equilibrium.

Lemma 1 shows that the Hessian of any game decomposes into symmetric and antisymmetric components. There are thus two 'pure' cases: when only the symmetric component is present, or only the antisymmetric. The first case, known as potential games (Monderer & Shapley, 1996), have been intensively studied because they are exactly the games where gradient descent *does* converge.

The second case, Hamiltonian[1] games, were not studied previously, probably because they coincide with zero-sum games in the bimatrix case (or constant-sum, depending on the constraints). Zero-sum and Hamiltonian games differ when the losses are not bilinear or there are more than two players. Hamiltonian games are important because (i) they are easy to solve and (ii) general games combine potential-like and Hamiltonian-like dynamics. The concept of a zero-sum game is too loose to be useful when there are many players: any n-player game can be reformulated as a zero-sum $(n + 1)$-player game where $\ell_{n+1} = -\sum_{i=1}^{n} \ell_i$. Zero-sum games are as complicated as general-sum games. Theorem 3 shows that Hamiltonian games obey a conservation law – which also provides the key to solving them, by gradient descent on the conserved quantity.

The general case, neither potential nor Hamiltonian, is more difficult and the focus of the remainder of the paper. Section 3 proposes *symplectic gradient adjustment (SGA)*, a gradient-based method for finding stable fixed points in general games. Appendix C contains TensorFlow code to compute the adjustment. The algorithm computes two Hessian-vector products, at a cost of two iterations of backprop. SGA satisfies a few natural desiderata: $D1$ it is compatible with the original dynamics and $D2$, $D3$ it is guaranteed to find stable equilibria in potential and Hamiltonian games.

For general games, correctly picking the *sign* of the adjustment (whether to add or subtract) is critical since it determines the behavior near stable and unstable equilibria. Section 2.3 defines stable equilibria and relates them to local Nash equilibria. Lemma 10 then shows how to set the sign so as to converge to stable fixed points. Correctly aligning

SGA allows higher learning rates and faster, more robust convergence, see theorem 7 and experiments in section 4.

Section 4 investigates a basic GAN setup from Metz et al. (2017), that tests for mode-collapse and mode hopping. Whereas simultaneous gradient descent completely fails; the symplectic adjustment leads to rapid convergence – slightly improved by correctly choosing the sign of the adjustment. Finally, section 3.5 applies the same criterion to align consensus optimization (Mescheder et al., 2017), preventing it from converging to unstable equilibria and (slightly) improving performance, figure 9 in the appendix.

Caveat. The behavior of SGA near fixed points that are neither negative nor positive semi-definite is not analysed. On the one hand, it was only recently shown that gradient descent behaves well near saddles when optimizing a single objective (Lee et al., 2016; 2017). On the other hand, Newton's method is attracted to saddles, see analysis and recently proposed remedy in Dauphin et al. (2014). Studying indefinite fixed points is deferred to future work.

Related work. Convergence to Nash equilibria in two-player games was studied in Singh et al. (2000). WoLF (Win or Learn Fast) converges to Nash equilibria in two-player two-action games (Bowling & Veloso, 2002). Extensions include weighted policy learning (Abdallah & Lesser, 2008) and GIGA-WoLF (Bowling, 2004). There has been almost no work on convergence to fixed points in general games. Optimistic mirror descent approximately converges in two-player bilinear zero-sum games (Daskalakis et al., 2018), a special case of Hamiltonian games. In more general settings it converges to coarse correlated equilibria.

There has been interesting recent work on convergence in GANs. Heusel et al. (2017) propose a two-time scale methods to find Nash. However, it likely scales badly with the number of players. Nagarajan & Kolter (2017) prove convergence for some algorithms, but under very strong assumptions Mescheder (2018). Consensus optimization (Mescheder et al., 2017) is discussed in section 3. Learning with opponent-learning awareness (LOLA) infinitesimally modifies the objectives of players to take into account their opponents' goals (Foerster et al., 2018). However, there are no guarantees that LOLA converges and in general it may modify the fixed-points of the game.

Notation. Dot products are written as $\mathbf{v}^{\mathsf{T}}\mathbf{w}$ or $\langle \mathbf{v}, \mathbf{w} \rangle$. The angle between vectors is $\theta(\mathbf{v}, \mathbf{w})$. Positive definiteness is denoted $\mathbf{S} \succ 0$. Omitted proofs are in appendix B.

2. The infinitesimal structure of games

In contrast to the classical formulation of games, we do not constrain the parameter sets (e.g. to the probability simplex) or require losses to be convex in the corresponding players'

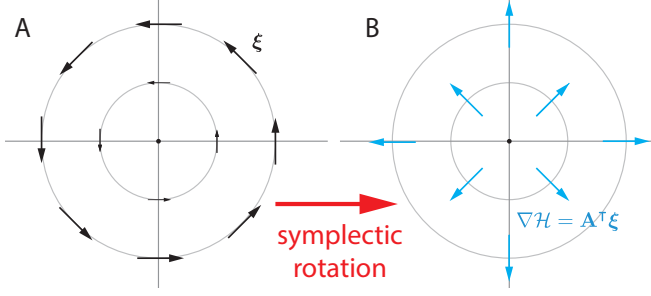[1] Lu (1992) defined an unrelated notion of Hamiltonian game.

Figure 1. **(A)** $\boldsymbol{\xi}$ cycles around the origin. **(B)** gradient descent on $\mathcal{H}$ converges to Nash.

parameters. Players could be interacting neural nets such as GANs (Goodfellow et al., 2014).

Definition 1. *A game is a set of players* $[n] = \{1, \ldots, n\}$ *and twice continuously differentiable losses* $\{\ell_i : \mathbb{R}^d \to \mathbb{R}\}_{i=1}^n$. *Parameters are* $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_n) \in \mathbb{R}^d$ *with* $\mathbf{w}_i \in \mathbb{R}^{d_i}$ *where* $\sum_{i=1}^n d_i = d$. *. Player i controls* $\mathbf{w}_i$.

It is sometimes convenient to write $\mathbf{w} = (\mathbf{w}_i, \mathbf{w}_{-i})$ where $\mathbf{w}_{-i}$ concatenates the parameters of all the players other than the ith, which is placed out of order by abuse of notation.

The *simultaneous gradient* is the gradient of the losses with respect to the parameters of the respective players:

$$\boldsymbol{\xi}(\mathbf{w}) = (\nabla_{\mathbf{w}_1}\ell_1, \ldots, \nabla_{\mathbf{w}_n}\ell_n) \in \mathbb{R}^d.$$

By the **dynamics** of the game, we mean following the negative of the vector field $\boldsymbol{\xi}$ with infinitesimal steps. There is no reason to expect $\boldsymbol{\xi}$ to be the gradient of a *single* function in general, and therefore no reason to expect the dynamics to converge to a fixed point.

2.1. Warmup: Hamiltonian mechanics in games

The next example illustrates the essential problem with gradients in games and the key insight motivating our approach.

Example 1 (Conservation of energy in a zero-sum unconstrained bimatrix game)**.** *Zero-sum games, where* $\sum_{i=1}^n \ell_i \equiv 0$, *are well-studied. The zero-sum game*

$$\ell_1(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\intercal \mathbf{A}\mathbf{y} \quad and \quad \ell_2(\mathbf{x}, \mathbf{y}) = -\mathbf{x}^\intercal \mathbf{A}\mathbf{y}$$

has Nash equilibrium at $(\mathbf{x}, \mathbf{y}) = (\mathbf{0}, \mathbf{0})$. *The simultaneous gradient* $\boldsymbol{\xi}(\mathbf{x}, \mathbf{y}) = (\mathbf{A}\mathbf{y}, -\mathbf{A}^\intercal \mathbf{x})$ *rotates around the Nash, see figure 1.*

The matrix A admits singular value decomposition (SVD) $\mathbf{A} = \mathbf{U}^\intercal \mathbf{D} \mathbf{V}$. *Changing to coordinates* $\mathbf{u} = \mathbf{D}^{\frac{1}{2}} \mathbf{U}\mathbf{x}$ *and* $\mathbf{v} = \mathbf{D}^{\frac{1}{2}} \mathbf{V}\mathbf{y}$ *gives* $\ell_{1/2}(\mathbf{u}, \mathbf{v}) = \pm \mathbf{u}^\intercal \mathbf{v}$. *Introduce the Hamiltonian*

$$\mathcal{H} = \frac{1}{2}\left(\|\mathbf{u}\|_2^2 + \|\mathbf{v}\|_2^2\right) = \frac{1}{2}\left(\mathbf{x}^\intercal \mathbf{U}^\intercal \mathbf{D}\mathbf{U}\mathbf{x} + \mathbf{y}^\intercal \mathbf{V}^\intercal \mathbf{D}\mathbf{V}\mathbf{y}\right).$$

Remarkably, the dynamics can be reformulated via Hamilton's equations in the coordinates given by the SVD of A:

$$\boldsymbol{\xi}(\mathbf{u}, \mathbf{v}) = \left(\frac{\partial \mathcal{H}}{\partial \mathbf{v}}, -\frac{\partial \mathcal{H}}{\partial \mathbf{u}}\right).$$

Vector field $\boldsymbol{\xi}$ cycles around the equilibrium because $\boldsymbol{\xi}$ conserves the Hamiltonian's level sets (i.e. $\langle \boldsymbol{\xi}, \nabla \mathcal{H} \rangle = 0$). *However,* **gradient descent on the Hamiltonian converges to the Nash equilibrium.** *The remainder of the paper explores the implications and limitations of this insight.*

Papadimitriou & Piliouras (2016) recently analyzed the dynamics of Matching Pennies (essentially, the above example) and showed with some effort that the cyclic behavior covers the entire parameter space. The Hamiltonian reformulation directly explains the cyclic behavior via a conservation law.

2.2. The generalized Helmholtz decomposition

The **Hessian** of a game is the $(d \times d)$-matrix of second-derivatives $\mathbf{H}(\mathbf{w}) := \nabla_{\mathbf{w}} \cdot \boldsymbol{\xi}(\mathbf{w})^\intercal = \left(\frac{\partial \xi_\alpha(\mathbf{w})}{\partial w_\beta}\right)_{\alpha,\beta=1}^d$. Concretely, the Hessian can be written

$$\mathbf{H}(\mathbf{w}) = \begin{pmatrix} \nabla^2_{\mathbf{w}_1}\ell_1 & \nabla^2_{\mathbf{w}_1,\mathbf{w}_2}\ell_1 & \cdots & \nabla^2_{\mathbf{w}_1,\mathbf{w}_n}\ell_1 \\ \nabla^2_{\mathbf{w}_2,\mathbf{w}_1}\ell_2 & \nabla^2_{\mathbf{w}_2}\ell_2 & \cdots & \nabla^2_{\mathbf{w}_2,\mathbf{w}_n}\ell_2 \\ \vdots & & & \vdots \\ \nabla^2_{\mathbf{w}_n,\mathbf{w}_1}\ell_n & \nabla^2_{\mathbf{w}_n,\mathbf{w}_2}\ell_n & \cdots & \nabla^2_{\mathbf{w}_n}\ell_n \end{pmatrix},$$

where $\nabla^2_{\mathbf{w}_i,\mathbf{w}_j}\ell_k$ is the $(d_i \times d_j)$-block of 2nd-order derivatives. The Hessian of a game is not necessarily symmetric. Note: α, β run over dimensions; i, j run over players.

Lemma 1 (generalized Helmholtz decomposition)**.** *The Hessian of any vector field decomposes uniquely into two components* $\mathbf{H}(\mathbf{w}) = \mathbf{S}(\mathbf{w}) + \mathbf{A}(\mathbf{w})$ *where* $\mathbf{S} \equiv \mathbf{S}^\intercal$ *is symmetric and* $\mathbf{A} + \mathbf{A}^\intercal \equiv 0$ *is antisymmetric.*

Proof. Any matrix decomposes uniquely as $\mathbf{M} = \mathbf{S} + \mathbf{A}$ where $\mathbf{S} = \frac{1}{2}(\mathbf{M} + \mathbf{M}^\intercal)$ and $\mathbf{A} = \frac{1}{2}(\mathbf{M} - \mathbf{M}^\intercal)$ are symmetric and antisymmetric. Applying the decomposition to the game Hessian yields the result. □

The decomposition is preserved by orthogonal change-of-coordinates $\mathbf{P}^\intercal \mathbf{M}\mathbf{P} = \mathbf{P}^\intercal \mathbf{S}\mathbf{P} + \mathbf{P}^\intercal \mathbf{A}\mathbf{P}$ since the terms remain symmetric and antisymmetric.

The connection to the classical Helmholtz decomposition in calculus is sketched in appendix E. Two obvious classes of games arise from the decomposition:

Definition 2. *A game is a **potential game** if* $\mathbf{A}(\mathbf{w}) \equiv 0$. *It is a **Hamiltonian game** if* $\mathbf{S}(\mathbf{w}) \equiv 0$.

Potential games are well-studied and easy to solve. Hamiltonian games are a new class of games that are also easy to solve. The general case is more delicate, see section 3.

2.3. Stable fixed points

Gradient-based methods can reliably find local – but not global – optima of nonconvex objective functions (Lee et al., 2016; 2017). Similarly, gradient-based methods cannot be expected to find global Nash equilibria in games.

Definition 3. *A fixed point* $\mathbf{w}^*$, *with* $\boldsymbol{\xi}(\mathbf{w}^*) = 0$, *is* **stable** *if* $\mathbf{S}(\mathbf{w}) \succeq 0$ *and* **unstable** *if* $\mathbf{S}(\mathbf{w}) \prec 0$ *for* $\mathbf{w}$ *in a neighborhood of* $\mathbf{w}^*$.

Fixed points that are neither positive nor negative definite are beyond the scope of the paper.

Lemma 2 (Stable fixed points are local Nash equilibria). *A point* $\mathbf{w}^*$ *is a* **local Nash equilibrium** *if, for all* i, *there is a* neighborhood U_i *of* $\mathbf{w}_i^*$ *such that* $\ell_i(\mathbf{w}_i, \mathbf{w}_{-i}^*) \geq \ell_i(\mathbf{w}_i^*, \mathbf{w}_{-i}^*)$ *for* $\mathbf{w}_i \in U_i$. *If fixed point* $\mathbf{w}^*$ *is stable then it is a local Nash equilibrium.*

Proof. If $\mathbf{S}$ is positive semidefinite then so are its $(d_i \times d_i)$-submatrices $\mathbf{S}_i := \nabla_{\mathbf{w}_i}^2 \ell_i$ for all i. The result follows. $\square$

Appendix A contains more details on local Nash equilibria.

2.4. Potential games

Potential games were introduced in Monderer & Shapley (1996). A game is a potential game if there is a single potential function $\phi : \mathbb{R}^d \to \mathbb{R}$ and positive numbers $\{\alpha_i > 0\}_{i=1}^n$ such that

$$\phi(\mathbf{w}_i', \mathbf{w}_{-i}) - \phi(\mathbf{w}_i'', \mathbf{w}_{-i}) = \alpha_i \Big(\ell_i(\mathbf{w}_i', \mathbf{w}_{-i}) - \ell_i(\mathbf{w}_i'', \mathbf{w}_{-i}) \Big)$$

for all i and all $\mathbf{w}_i', \mathbf{w}_i'', \mathbf{w}_{-i}$. Monderer & Shapley (1996) show a game is a potential game iff $\alpha_i \nabla_{\mathbf{w}_i} \ell_i = \nabla_{\mathbf{w}_i} \phi$ for all i, which is equivalent to

$$\alpha_i \nabla_{\mathbf{w}_i \mathbf{w}_j}^2 \ell_i = \alpha_j \nabla_{\mathbf{w}_i \mathbf{w}_j}^2 \ell_j = \alpha_j \left(\nabla_{\mathbf{w}_j \mathbf{w}_i}^2 \ell_j \right)^{\mathsf{T}} \quad \forall i, j.$$

Our definition of potential game is the special case where $\alpha_i = 1$ for all i, which Monderer & Shapley (1996) call an **exact** potential game. We use the shorthand 'potential game' to refer to exact potential games in what follows.

Potential games have been extensively studied since they are one of the few classes of games for which Nash equilibria can be computed (Rosenthal, 1973). For our purposes, they are games where simultaneous gradient descent on the losses is gradient descent on a single function. It follows that descent on $\boldsymbol{\xi}$ converges to a fixed point that is a local minimum of ϕ – or saddle, but see Lee et al. (2017).

2.5. Hamiltonian games

A concrete example may help understand antisymmetric matrices. Suppose n competitors play one-on-one and that the

probability of player i beating player j is p_{ij}. Then, assuming there are no draws, the probabilities satisfy $p_{ij} + p_{ji} = 1$ and $p_{ii} = \frac{1}{2}$. The matrix $\mathbf{A} = \left(\log \frac{p_{ij}}{1 - p_{ij}} \right)_{i,j=1}^n$ of logits is then antisymmetric. Intuitively, antisymmetry reflects a *hyperadversarial* setting where all pairwise interactions between players are zero-sum. In general, Hamiltonian games are related to – but distinct from – zero-sum games.

Example 2 (an unconstrained[2] bimatrix game is zero–sum iff it is Hamiltonian). *Consider bimatrix game with* $\ell_1(\mathbf{x}, \mathbf{y}) = \mathbf{x}^{\mathsf{T}} \mathbf{P} \mathbf{y}$ *and* $\ell_2(\mathbf{x}, \mathbf{y}) = \mathbf{x}^{\mathsf{T}} \mathbf{Q} \mathbf{y}$. *Then* $\boldsymbol{\xi} = (\mathbf{P} \mathbf{y}, \mathbf{Q}^{\mathsf{T}} \mathbf{x})$ *and the Hessian components have block structure*

$$2\mathbf{A} = \begin{pmatrix} 0 & \mathbf{P} - \mathbf{Q} \\ (\mathbf{Q} - \mathbf{P})^{\mathsf{T}} & 0 \end{pmatrix} \quad 2\mathbf{S} = \begin{pmatrix} 0 & \mathbf{P} + \mathbf{Q} \\ (\mathbf{P} + \mathbf{Q})^{\mathsf{T}} & 0 \end{pmatrix}$$

The game is Hamiltonian iff $\mathbf{S} = 0$ *iff* $\mathbf{P} + \mathbf{Q} = 0$ *iff* $\ell_1 + \ell_2 = 0$.

There are Hamiltonian games that are not zero-sum and vice versa.

Example 3 (Hamiltonian game that is not zero-sum). *Fix constants* a *and* b *and suppose players* 1 *and* 2 *minimize losses*

$$\ell_1(x, y) = x(y - b) \quad \text{and} \quad \ell_2(x, y) = -(x - a)y$$

with respect to x *and* y *respectively.*

Example 4 (zero-sum game that is not Hamiltonian). *Players 1 and 2 minimize*

$$\ell_1(x, y) = x^2 + y^2 \qquad \ell_2(x, y) = -(x^2 + y^2).$$

The game actually has potential function $\phi(x, y) = x^2 - y^2$.

Hamiltonian games are quite different from potential games. There is a Hamiltonian function $\mathcal{H}$ that specifies a conserved quantity. Whereas the dynamics *equal* $\nabla \phi$ in potential games; they are *orthogonal* to $\nabla \mathcal{H}$ in Hamiltonian games. The orthogonality implies the conservation law that underlies the cyclic behavior in example 1.

Theorem 3. *Let* $\mathcal{H}(\mathbf{w}) := \frac{1}{2} \|\boldsymbol{\xi}(\mathbf{w})\|_2^2$. *If the game is Hamiltonian then (i)* $\nabla \mathcal{H} = \mathbf{A}^{\mathsf{T}} \boldsymbol{\xi}$ *and (ii)* $\boldsymbol{\xi}$ **preserves the level sets** *of* $\mathcal{H}$ *since* $\langle \boldsymbol{\xi}, \nabla \mathcal{H} \rangle = 0$. *If the Hessian is invertible and* $\lim_{\|\mathbf{w}\| \to \infty} \mathcal{H}(\mathbf{w}) = \infty$ *then (iii) gradient descent on* $\mathcal{H}$ *converges to a local Nash equilibrium.*

In fact, $\mathcal{H}$ is a Hamiltonian[3] function for the game dynamics. We use the notation $\mathcal{H}(\mathbf{w}) = \frac{1}{2} \|\boldsymbol{\xi}(\mathbf{w})\|^2$ throughout the paper. However, $\mathcal{H}$ is only a Hamiltonian function for $\boldsymbol{\xi}$ if the game is Hamiltonian.

---

[2]The parameters in bimatrix games are usually constrained to the probability simplex.

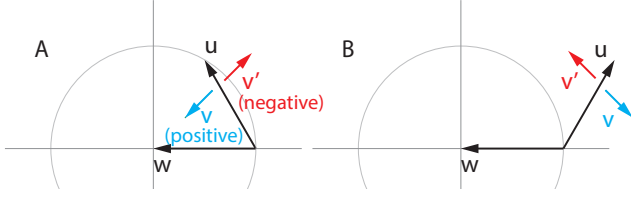[3]Notation: $\mathcal{H}$ for Hamiltonian; $\mathbf{H}$ for Hessian.

Figure 2. Infinitesimal alignment is positive (cyan) when small positive λ either: **(A)** pulls **u** toward **w**, if **w** and **u** have angle $< 90°$; or **(B)** pushes **u** away from **w** if their angle is $> 90°$.

There is a precise mapping from Hamiltonian games to symplectic geometry, see Appendix E. Symplectic geometry is the modern formulation of classical mechanics. Recall that periodic behaviors (e.g. orbits) often arise in classical mechanics. The orbits lie on the level sets of the Hamiltonian, which expresses the total energy of the system.

3. Algorithms

Fixed points of potential and Hamiltonian games can be found by descent on $\boldsymbol{\xi}$ and $\nabla\mathcal{H}$ respectively. This section tackles finding stable fixed points in general games.

3.1. Finding fixed points

If the Hessian $\mathbf{H}(\mathbf{w})$ is invertible, then $\nabla\mathcal{H} = \mathbf{H}^\mathsf{T}\boldsymbol{\xi} = 0$ iff $\boldsymbol{\xi} = 0$. Thus, gradient descent on $\mathcal{H}$ converges to fixed points of $\boldsymbol{\xi}$. However, there is no guarantee that descent on $\mathcal{H}$ will find a *stable* fixed point.

Mescheder et al. (2017) propose *consensus optimization*, a gradient adjustment of the form

$$\boldsymbol{\xi} + \lambda \cdot \mathbf{H}^\mathsf{T}\boldsymbol{\xi} = \boldsymbol{\xi} + \lambda \cdot \nabla\mathcal{H}.$$

Unfortunately, consensus optimization can converge to unstable fixed points even in simple cases where the 'game' is to minimize a single function:

Example 5 (consensus optimization converges to global maximum). *Consider a potential game with losses $\ell_1(x, y) = \ell_2(x, y) = -\frac{\kappa}{2}(x^2 + y^2)$ with $\kappa \gg 0$. Then*

$$\boldsymbol{\xi} = -\kappa \cdot \begin{pmatrix} x \\ y \end{pmatrix} \quad and \quad \mathbf{H} = -\begin{pmatrix} \kappa & 0 \\ 0 & \kappa \end{pmatrix}$$

Note that $\|\boldsymbol{\xi}\|^2 = \kappa^2(x^2 + y^2)$ and

$$\boldsymbol{\xi} + \lambda \cdot \mathbf{H}^\mathsf{T}\boldsymbol{\xi} = \kappa(\lambda\kappa - 1) \cdot \begin{pmatrix} x \\ y \end{pmatrix}.$$

Descent on $\boldsymbol{\xi} + \lambda \cdot \mathbf{H}^\mathsf{T}\boldsymbol{\xi}$ converges to the global maximum $(x, y) = (0, 0)$ unless $\lambda < \frac{1}{\kappa}$.

Although consensus optimization works well in some special cases like two-player zero-sum; it cannot be considered a candidate algorithm for finding stable fixed points in general games, since it fails in the basic case of potential games.

Algorithm 1 Symplectic Gradient Adjustment

> **Input:** losses $\mathcal{L} = \{\ell_i\}_{i=1}^n$, weights $\mathcal{W} = \{\mathbf{w}_i\}_{i=1}^n$
> $\boldsymbol{\xi} \leftarrow \left[\texttt{gradient}(\ell_i, \mathbf{w}_i) \text{ for } (\ell_i, \mathbf{w}_i) \in (\mathcal{L}, \mathcal{W})\right]$
> $\mathbf{A}^\mathsf{T}\boldsymbol{\xi} \leftarrow \texttt{get\_sym\_adj}(\mathcal{L}, \mathcal{W})$ // appendix C
> **if** $\texttt{align}$ **then**
> $\nabla\mathcal{H} \leftarrow \left[\texttt{gradient}(\frac{1}{2}\|\boldsymbol{\xi}\|^2, \mathbf{w}) \text{ for } \mathbf{w} \in \mathcal{W}\right]$
> $\lambda \leftarrow \texttt{sign}\left(\frac{1}{d}\langle\boldsymbol{\xi}, \nabla\mathcal{H}\rangle\langle\mathbf{A}^\mathsf{T}\boldsymbol{\xi}, \nabla\mathcal{H}\rangle + \epsilon\right)$ // $\epsilon = \frac{1}{10}$
> **else**
> $\lambda \leftarrow 1$
> **end if**
> **Output:** $\boldsymbol{\xi} + \lambda \cdot \mathbf{A}^\mathsf{T}\boldsymbol{\xi}$ // plug into any optimizer

---

3.2. Finding stable fixed points

There are two classes of games where we know how to find stable fixed points: potential games where $\boldsymbol{\xi}$ converges to a local minimum and Hamiltonian games where $\nabla\mathcal{H}$, which is orthogonal to $\boldsymbol{\xi}$, finds stable fixed points. In the general case, the following desiderata provide a set of reasonable properties for an adjustment $\boldsymbol{\xi}_\lambda$ of the game dynamics:

Desiderata. *To find stable fixed points, an adjustment $\boldsymbol{\xi}_\lambda$ to the game dynamics should satisfy*

D1. compatible with game dynamics: $\langle\boldsymbol{\xi}_\lambda, \boldsymbol{\xi}\rangle = \alpha_1 \cdot \|\boldsymbol{\xi}\|^2$;

D2. compatible with potential dynamics: *if the game is a potential game then* $\langle\boldsymbol{\xi}_\lambda, \nabla\phi\rangle = \alpha_2 \cdot \|\nabla\phi\|^2$;

D3. compatible with Hamiltonian dynamics: *If the game is Hamiltonian then* $\langle\boldsymbol{\xi}_\lambda, \nabla\mathcal{H}\rangle = \alpha_3 \cdot \|\nabla\mathcal{H}\|^2$;

D4. attracted to stable equilibria: *in neighborhoods where* $\mathbf{S} \succ 0$, *require* $\theta(\boldsymbol{\xi}_\lambda, \nabla\mathcal{H}) \leq \theta(\boldsymbol{\xi}, \nabla\mathcal{H})$;

D5. repelled by unstable equilibria: *in neighborhoods where* $\mathbf{S} \prec 0$, *require* $\theta(\boldsymbol{\xi}_\lambda, \nabla\mathcal{H}) \geq \theta(\boldsymbol{\xi}, \nabla\mathcal{H})$;

for some $\alpha_1, \alpha_2, \alpha_3 > 0$.

Desideratum $D1$ does not guarantee that players act in their own self-interest – this requires a stronger positivity condition on dot-products with subvectors of $\boldsymbol{\xi}$, see Balduzzi (2017). Desiderata $D4$ and $D5$ are explained in section 3.4. The unadjusted dynamics $\boldsymbol{\xi}$ satisfies all the desiderata except $D3$. Consensus optimization only satisfies $D3$ and $D4$. Ideally, desideratum $D5$ would be strengthened to 'repelled by saddle points where $\boldsymbol{\xi}(\mathbf{w}) = 0$ but $\mathbf{S}(\mathbf{w}) \not\succeq 0$'.

3.3. Symplectic gradient adjustment

Proposition 4. *The **symplectic gradient adjustment (SGA)***

$$\boldsymbol{\xi}_\lambda := \boldsymbol{\xi} + \lambda \cdot \mathbf{A}^\mathsf{T}\boldsymbol{\xi}.$$

satisfies $D1$–$D3$ for $\lambda > 0$, with $\alpha_1 = 1 = \alpha_2$ and $\alpha_3 = \lambda$.

Note that desiderata $D1$ and $D2$ are true even when $\lambda < 0$. This will prove useful, since example 6 and theorem 5 show it is necessary pick negative λ near $\mathbf{S} \prec 0$. Section 3.4 shows how to also satisfy desiderata $D4$ and $D5$.

Proof. First claim: $\lambda \cdot \boldsymbol{\xi}^{\mathsf{T}} \mathbf{A}^{\mathsf{T}} \boldsymbol{\xi} = 0$ by skew-symmetry of $\mathbf{A}$. Second claim: $\mathbf{A} \equiv 0$ in a potential game, so $\boldsymbol{\xi}_\lambda = \boldsymbol{\xi} = \nabla \phi$. Third claim: $\langle \boldsymbol{\xi}_\lambda, \nabla \mathcal{H} \rangle = \langle \boldsymbol{\xi}_\lambda, \mathbf{H}^{\mathsf{T}} \boldsymbol{\xi} \rangle = \langle \boldsymbol{\xi}_\lambda, \mathbf{A}^{\mathsf{T}} \boldsymbol{\xi} \rangle = \lambda \cdot \boldsymbol{\xi}^{\mathsf{T}} \mathbf{A} \mathbf{A}^{\mathsf{T}} \boldsymbol{\xi} = \lambda \cdot \|\nabla \mathcal{H}\|^2$ since $\mathbf{H} = \mathbf{A}$ by assumption and since $\boldsymbol{\xi}^{\mathsf{T}} \mathbf{A}^{\mathsf{T}} \boldsymbol{\xi} = 0$ by antisymmetry. $\square$

In the example below, almost any choice of positive λ results in convergence to an unstable equilibrium. The problem arises from the combination of a weak repellor with a strong rotational force. The next section shows how to pick λ to avoid unstable equilibria.

Example 6 (failure case for $\lambda > 0$). *Suppose $\epsilon > 0$ is small and*

$$f(x,y) = -\frac{\epsilon}{2}x^2 - xy \ \text{ and } \ g(x,y) = -\frac{\epsilon}{2}y^2 + xy$$

with an unstable equilibrium at $(0,0)$. The dynamics are

$$\boldsymbol{\xi} = \epsilon \cdot \begin{pmatrix} -x \\ -y \end{pmatrix} + \begin{pmatrix} -y \\ x \end{pmatrix} \quad \text{with} \quad \mathbf{A} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

and

$$\mathbf{A}^{\mathsf{T}} \boldsymbol{\xi} = \begin{pmatrix} x \\ y \end{pmatrix} + \epsilon \begin{pmatrix} -y \\ x \end{pmatrix}$$

Finally observe that

$$\boldsymbol{\xi} + \lambda \cdot \mathbf{A}^{\mathsf{T}} \boldsymbol{\xi} = (\lambda - \epsilon) \cdot \begin{pmatrix} x \\ y \end{pmatrix} + (1 + \epsilon\lambda) \cdot \begin{pmatrix} -y \\ x \end{pmatrix}$$

which converges to the unstable equilibrium if $\lambda > \epsilon$.

Lemma 9 in the appendix shows that, if $\mathbf{S}$ and $\mathbf{A}$ commute and $\mathbf{S} \succeq 0$, then $\langle \boldsymbol{\xi}_\lambda, \nabla \mathcal{H} \rangle \geq 0$ for $\lambda > 0$. The proof of theorem 5 introduces the additive condition number to upper-bound the worst-case noncommutativity of $\mathbf{S}$, which allows to quantify the relationship between $\boldsymbol{\xi}_\lambda$ and $\nabla \mathcal{H}$.

Theorem 5. *Let $\mathbf{S}$ be a symmetric matrix with eigenvalues $\sigma_{max} \geq \cdots \geq \sigma_{min}$. The **additive condition number**[4] of $\mathbf{S}$ is $\kappa = \sigma_{max} - \sigma_{min}$. If $\mathbf{S} \succeq 0$ is positive semidefinite with additive condition number κ then $\lambda \in (0, \frac{4}{\kappa})$ implies*

$$\langle \boldsymbol{\xi}_\lambda, \nabla \mathcal{H} \rangle \geq 0.$$

If $\mathbf{S}$ is negative semidefinite, then $\lambda \in (0, \frac{4}{\kappa})$ implies

$$\langle \boldsymbol{\xi}_{-\lambda}, \nabla \mathcal{H} \rangle \leq 0.$$

The inequalities are strict if $\mathbf{H}$ is invertible.

If $\kappa = 0$, then $\mathbf{S} = \sigma \cdot \mathbf{I}$ commutes with *all* matrices. The larger the additive condition number κ, the larger the *potential* failure of $\mathbf{S}$ to commute with other matrices.

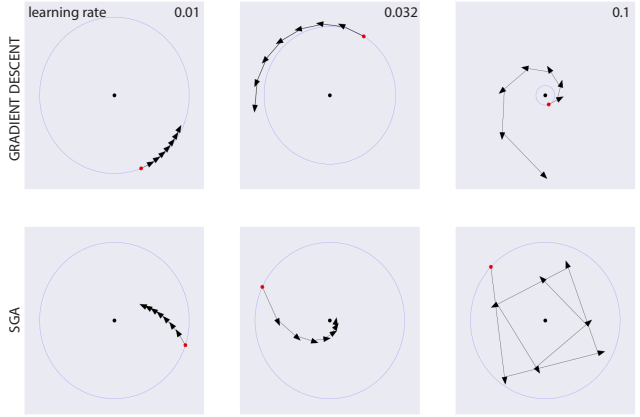[4] The condition number of a positive definite matrix is $\frac{\sigma_{max}}{\sigma_{min}}$.

Figure 3. SGA allows faster and more robust convergence to stable fixed points. Note the scale of top-right panel differs from rest.

3.4. How to pick $\text{sign}(\lambda)$

This section explains desiderata $D4$–$D5$ and shows how to pick $\text{sign}(\lambda)$ to speed up convergence towards stable and away from unstable fixed points.

First, observe that $\langle \boldsymbol{\xi}, \nabla \mathcal{H} \rangle = \boldsymbol{\xi}^{\mathsf{T}} (\mathbf{S} + \mathbf{A})^{\mathsf{T}} \boldsymbol{\xi} = \boldsymbol{\xi}^{\mathsf{T}} \mathbf{S} \boldsymbol{\xi}$. It follows that for $\boldsymbol{\xi} \neq 0$:

$$\begin{cases} \text{if } \mathbf{S} \succeq 0 & \text{then } \langle \boldsymbol{\xi}, \nabla \mathcal{H} \rangle \geq 0; \\ \text{if } \mathbf{S} \prec 0 & \text{then } \langle \boldsymbol{\xi}, \nabla \mathcal{H} \rangle < 0. \end{cases} \quad (1)$$

A criterion to probe the positive/negative definiteness of $\mathbf{S}$ is thus to check the sign of $\langle \boldsymbol{\xi}, \nabla \mathcal{H} \rangle$. The dot product can take any value if $\mathbf{S}$ is not positive nor negative definite. The behavior near saddles is beyond the scope of the paper.

Desiderata $D4$ can be interpreted as follows. If $\boldsymbol{\xi}$ points at a stable equilibrium then we require that $\boldsymbol{\xi}_\lambda$ points *more* towards the equilibrium (i.e. has smaller angle). Conversely if $\boldsymbol{\xi}$ points away then the adjustment should point *further* away. More formally,

Definition 4. *Let $\mathbf{u}$ and $\mathbf{v}$ be two vectors. The **infinitesimal alignment** of $\boldsymbol{\xi}_\lambda := \mathbf{u} + \lambda \cdot \mathbf{v}$ with a third vector $\mathbf{w}$ is*

$$\text{align}(\boldsymbol{\xi}_\lambda, \mathbf{w}) := \frac{d}{d\lambda} \left\{ \cos^2 \theta_\lambda \right\}_{|\lambda=0} \ \text{for } \theta_\lambda := \theta(\boldsymbol{\xi}_\lambda, \mathbf{w}).$$

If $\mathbf{u}$ and $\mathbf{w}$ point the same way, $\mathbf{u}^{\mathsf{T}} \mathbf{w} > 0$, then $\text{align} > 0$ when $\mathbf{v}$ bends $\mathbf{u}$ further toward $\mathbf{w}$, see figure 2A. Otherwise $\text{align} > 0$ when $\mathbf{v}$ bends $\mathbf{u}$ away from $\mathbf{w}$, figure 2B.

Proposition 6. *Desiderata $D4$–$D5$ are satisfied for λ such that $\lambda \cdot \langle \boldsymbol{\xi}, \nabla \mathcal{H} \rangle \cdot \langle \mathbf{A}^{\mathsf{T}} \boldsymbol{\xi}, \nabla \mathcal{H} \rangle \geq 0$.*

Computing the sign of $\langle \boldsymbol{\xi}, \nabla \mathcal{H} \rangle$ provides a check for stable and unstable fixed points. Computing the sign of $\langle \mathbf{A}^{\mathsf{T}} \boldsymbol{\xi}, \nabla \mathcal{H} \rangle$ checks whether the adjustment term points towards or away from the (nearby) fixed point. Putting the two checks together yields a prescription for the sign of λ.
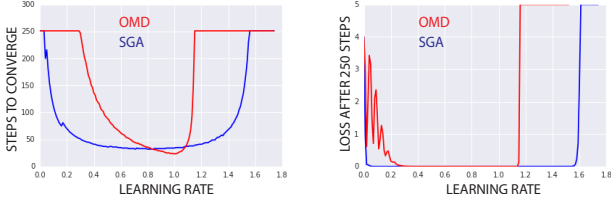
Figure 4. Comparison of SGA with optimistic mirror descent. Sweep over learning rates in range $[0.01, 1.75]$; $\lambda = 1$ throughout. **(Left):** iterations to convergence (up to 250). **(Right):** average absolute value of losses over iterations 240-250, cutoff at 5.

Alignment and convergence rates. Finally, we show that increasing alignment helps speed convergence. Gradient descent is also known as the method of steepest descent. In general games, however, $\boldsymbol{\xi}$ does not follow the steepest path to fixed points due to the 'rotational force', which forces lower learning rates and slows down convergence:

Theorem 7. *Suppose f is convex and Lipschitz smooth with $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \cdot \|\mathbf{x} - \mathbf{y}\|$. Let $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \cdot \mathbf{v}$ where $\|\mathbf{v}\| = \|\nabla f(\mathbf{w}_t)\|$. Then the optimal step size is $\eta^* = \frac{\cos \theta}{L}$ where $\theta := \theta(\nabla f(\mathbf{w}_t), \mathbf{v})$, with*

$$f(\mathbf{w}_{t+1}) \leq f(\mathbf{w}_t) - \frac{\cos^2 \theta}{2L} \cdot \|\nabla f(\mathbf{w}_t)\|^2.$$

Increasing the cosine with the steepest direction improves convergence. The alignment computation in algorithm 1 chooses λ to be positive or negative such that $\boldsymbol{\xi}_\lambda$ is bent towards stable (increasing the cosine) and away from unstable fixed points. Adding a small $\epsilon > 0$ to the computation introduces a weak bias towards stable fixed points.

3.5. Aligned consensus optimization

The stability criterion in (1) also provides a simple way to prevent consensus optimization from converging to unstable equilibria. **Aligned consensus optimization** is

$$\boldsymbol{\xi} + |\lambda| \cdot \text{sign}\left(\langle \boldsymbol{\xi}, \nabla \mathcal{H} \rangle\right) \cdot \mathbf{H}^\intercal \boldsymbol{\xi}, \qquad (2)$$

where in practice we set $\lambda = 1$. Aligned consensus satisfies desiderata *D3–D5*. However, it behaves strangely in potential games. Multiplying by the Hessian is the 'inverse' of Newton's method: it increases the gap between small and large eigenvalues, increasing the (usual, multiplicative) condition number and slows down convergence. Nevertheless, consensus optimization works well in GANs (Mescheder et al., 2017), and aligned consensus may improve performance, see figure 9 in appendix.

Finally, note that dropping the first term $\boldsymbol{\xi}$ from (2) yields a simpler update that also satisfies *D3–D5*. However, the resulting algorithm performs poorly in experiments (not shown), perhaps because it is attracted to saddles.

4. Experiments

We compare SGA with simultaneous gradient descent, optimistic mirror descent (Daskalakis et al., 2018) and consensus optimization (Mescheder et al., 2017) in basic settings.

4.1. Learning rates and alignment

We investigate the effect of SGA when a weak attractor is coupled to a strong rotational force:

$$\ell_1(x, y) = \frac{1}{2}x^2 + 10xy \quad \text{and} \quad \ell_2(x, y) = \frac{1}{2}y^2 - 10xy$$

Gradient descent is extremely sensitive to the choice of learning rate η, top row of figure 3. As η increases through $\{0.01, 0.032, 0.1\}$ gradient descent goes from converging extremely slowly, to diverging slowly, to diverging rapidly. SGA yields faster, more robust convergence. SGA converges faster with learning rates $\eta = 0.01$ and $\eta = 0.032$, and only starts overshooting the fixed point for $\eta = 0.1$.

4.2. Basic adversarial games

Figure 4 compares SGA with optimistic mirror descent on a zero-sum bimatrix game with $\ell_{1/2}(\mathbf{w}_1, \mathbf{w}_2) = \pm \mathbf{w}_1^\intercal \mathbf{w}_2$. The example is modified from Daskalakis et al. (2018) who also consider a linear offset that makes no difference. A run converges, panel A, if the average absolute value of losses on the last 10 iterations is < 0.01.

Although OMD's peak performance is better than SGA, we find that SGA converges – and does so faster – for a wider range of learning rates. OMD diverges for learning rates not in the range [0.3, 1.2]. Simultaneous gradient descent oscillates without converging (not shown). Individual runs are shown in the appendix. The appendix also compares the performance of the algorithms on four-player games.

4.3. Generative adversarial networks

We apply SGA to a basic setup adapted from Metz et al. (2017). Data is sampled from a highly multimodal distribution designed to probe the tendency to collapse onto a subset of modes during training. The distribution is a mixture of 16 Gaussians arranged in a 4×4 grid, see ground truth in figure 8. The generator and discriminator networks both have 6 ReLU layers of 384 neurons. The generator has two output neurons; the discriminator has one.

Figure 5 shows results after $\{2000, 4000, 6000, 8000\}$ iterations. The networks are trained under RMSProp. Learning rates were chosen by visual inspection of grid search results at iteration 8000, see appendix. Simultaneous gradient descent and SGA are shown in the figure. Results for consensus optimization are in the appendix.

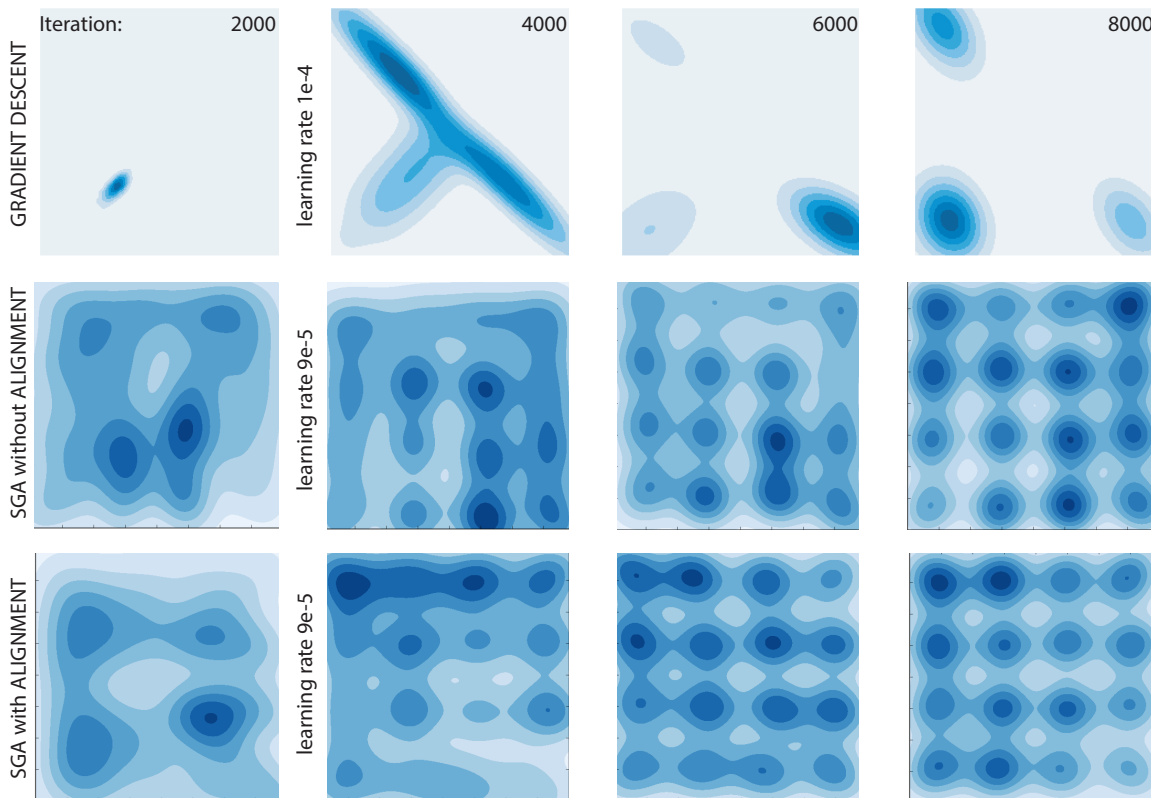Simultaneous gradient descent exhibits mode collapse fol-

Figure 5. **Top:** Simultaneous gradient descent suffers from mode collapse and in later iterations (not shown) mode hopping. **Middle:** vanilla SGA converges smoothly to the ground truth (figure 8 in appendix). **Bottom:** SGA with alignment converges slightly faster.

lowed by mode hopping in later iterations (not shown). Mode hopping is analogous to the cycles in example 1. Unaligned SGA converges to the correct distribution; alignment speeds up convergence slightly. Consensus optimization performs similarly, see figure 9 in the appendix. However, it can converge to local maxima, recall example 5.

5. Discussion

Modern deep learning treats differentiable modules like plug-and-play lego blocks. For this to work, at the very least, we need to know that gradient descent will find local minima. Unfortunately, gradient descent does *not* necessarily find local minima when optimizing multiple interacting objectives. With the recent proliferation of algorithms that optimize more than one loss, it is becoming increasingly urgent to understand and control the dynamics of interacting losses. Although there is interesting recent work on two-player adversarial games such as GANs, there is essentially no work on finding stable fixed points in general.

The generalized Helmholtz decomposition provides a powerful new perspective on game dynamics. A key feature is that the analysis is indifferent to the number of players. Instead, it is the interplay between the simultaneous gradient ξ on the losses and the symmetric and antisymmetric

matrices of second-order terms that guides algorithm design and governs the dynamics under gradient adjustments.

Symplectic gradient adjustment is a straightforward application of the Helmholtz decomposition. It is unlikely that SGA is the only or best approach to finding stable fixed points. A deeper understanding of the interaction between the potential and Hamiltonian components will lead to more effective algorithms. Of particular interest are pure first-order methods that do not use Hessian-vector products.

Finally, it is worth raising a philosophical point. The goal in this paper is to find stable fixed points (e.g. because in GANs they yield pleasing samples). We are not concerned with the losses of the players *per se*. The gradient adjustments may lead to a player acting against its own self-interest by increasing its loss. We consider this acceptable insofar as it encourages convergence to a stable fixed point.

References

Abdallah, S. and Lesser, V. R. A multiagent reinforcement learning algorithm with non-linear dynamics. *J. Artif.*

Intell. Res., 33:521–549, 2008.

Balduzzi, D. Strongly-Typed Agents are Guaranteed to Interact Safely. In *ICML*, 2017.

Bottacin, F. A Marsden-Weinstein Reduction Theorem for Presymplectic Manifolds. 2005.

Bowling, M. and Veloso, M. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

Bowling, M. H. Convergence and no-regret in multiagent learning. In *NIPS*, pp. 209–216, 2004.

Candogan, O., Menache, I., Ozdaglar, A., and Parrilo, P. A. Flows and decompositions of games: Harmonic and potential games. *Mathematics of Operations Research*, 36 (3):474–503, 2011.

Daskalakis, C., Goldberg, P. W., and Papadimitriou, C. The Complexity of Computing a Nash Equilibrium. *SIAM J. Computing*, 39(1):195–259, 2009.

Daskalakis, C., Ilyas, A., Syrgkanis, V., and Zeng, H. Training GANs with Optimism. In *ICLR*, 2018.

Dauphin, Y., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *NIPS*, 2014.

Foerster, J. N., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. Learning with Opponent-Learning Awareness. In *AAMAS*, 2018.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Nets. In *NIPS*, 2014.

Hart, S. and Mas-Colell, A. *Simple Adaptive Strategies: From Regret-Matching to Uncoupled Dynamics*. World Scientific, 2013.

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. In *NIPS*, 2017.

Hu, J. and Wellman, M. P. Nash Q-Learning for General-Sum Stochastic Games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., and Kavukcuoglu, K. Decoupled Neural Interfaces using Synthetic Gradients. In *ICML*, 2017.

Lee, J., Panageas, I., Piliouras, G., Simchowitz, M., Jordan, M., and Recht, B. First-order Methods Almost Always Avoid Saddle Points. In *arXiv:1710.07406*, 2017.

Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient Descent Converges to Minimizers. In *COLT*, 2016.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. Proximal Gradient Temporal Difference Learning Algorithms. In *IJCAI*, 2016.

Lu, X. Hamiltonian games. *Journal of Combinatorial Theory, Series B*, 55:18–32, 1992.

Mertikopoulos, P., Papadimitriou, C., and Piliouras, G. Cycles in adversarial regularized learning. In *SODA*, 2018.

Mescheder, L. On the convergence properties of GAN training. In *ArXiv:1801:04406*, 2018.

Mescheder, L., Nowozin, S., and Geiger, A. The Numerics of GANs. In *NIPS*. 2017.

Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. Unrolled generative adversarial networks. In *ICLR*, 2017.

Monderer, D. and Shapley, L. S. Potential Games. *Games and Economic Behavior*, 14:124–143, 1996.

Nagarajan, V. and Kolter, J. Z. Gradient descent GAN optimization is locally stable. In *NIPS*, 2017.

Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, 2004.

Papadimitriou, C. and Piliouras, G. From Nash Equilibria to Chain Recurrent Sets: Solution Concepts and Topology. In *ITCS*, 2016.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven Exploration by Self-supervised Prediction. In *ICML*, 2017.

Pfau, D. and Vinyals, O. Connecting Generative Adversarial Networks and Actor-Critic Methods. In *arXiv:1610.01945*, 2016.

Racanière, S., Weber, T., Reichert, D. P., Buesing, L., Guez, A., Rezende, D. J., Badia, A. P., Vinyals, O., Heess, N., Li, Y., Pascanu, R., Battaglia, P., Hassabis, D., Silver, D., and Wierstra, D. Imagination-Augmented Agents for Deep Reinforcement Learning. In *NIPS*, 2017.

Rosenthal, R. W. A Class of Games Possessing Pure-Strategy Nash Equilibria. *Int J Game Theory*, 2:65–67, 1973.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved Techniques for Training GANs. In *NIPS*, 2016.

Shoham, Y. and Leyton-Brown, K. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

Singh, S., Kearns, M., and Mansour, Y. Nash Convergence of Gradient Dynamics in General-Sum Games. In *UAI*, 2000.

Stoltz, G. and Lugosi, G. Learning correlated equilibria in games with compact sets of strategies. *Games and Economic Behavior*, 59:187–208, 2007.

Vezhnevets, A., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. FeUdal Networks for Hierarchical Reinforcement Learning. In *ICML*, 2017.

Wayne, G. and Abbott, L. F. Hierarchical Control Using Networks Trained with Higher-Level Forward Models. *Neural Computation*, (26), 2014.