Sample-based Approximation of Nash in Large Many-Player Games via Gradient Descent

Ian Gemp
DeepMind
London, United Kingdom
imgemp@deepmind.com

Rahul Savani
University of Liverpool
Liverpool, United Kingdom
rahul.savani@liverpool.ac.uk

Marc Lanctot
DeepMind
Edmonton, Canada
lanctot@deepmind.com

Yoram Bachrach
DeepMind
London, United Kingdom
yorambac@deepmind.com

Thomas Anthony
DeepMind
London, United Kingdom
twa@deepmind.com

Richard Everett
DeepMind
London, United Kingdom
reverett@deepmind.com

Andrea Tacchetti
DeepMind
London, United Kingdom
atacchet@deepmind.com

Tom Eccles
DeepMind
London, United Kingdom
eccles@deepmind.com

János Kramár
DeepMind
London, United Kingdom
janosk@deepmind.com

ABSTRACT

Nash equilibrium is a central concept in game theory. Several Nash solvers exist, yet none scale to normal-form games with many actions and many players, especially those with payoff tensors too big to be stored in memory. In this work, we propose an approach that iteratively improves an approximation to a Nash equilibrium through joint play. It accomplishes this by tracing a previously established homotopy that defines a continuum of equilibria for the game regularized with decaying levels of entropy. This continuum asymptotically approaches the *limiting logit equilibrium*, proven by McKelvey and Palfrey (1995) to be unique in *almost* all games, thereby partially circumventing the well-known equilibrium selection problem of many-player games. To encourage iterates to remain near this path, we efficiently minimize *average deviation incentive* via stochastic gradient descent, intelligently sampling entries in the payoff tensor as needed. Monte Carlo estimates of the stochastic gradient from joint play are biased due to the appearance of a nonlinear max operator in the objective, so we introduce additional innovations to the algorithm to alleviate gradient bias. The descent process can also be viewed as repeatedly constructing and reacting to a polymatrix approximation to the game. In these ways, our proposed approach, *average deviation incentive descent with adaptive sampling* (ADIDAS), is most similar to three classical approaches, namely homotopy-type, Lyapunov, and iterative polymatrix solvers. The lack of local convergence guarantees for biased gradient descent prevents guaranteed convergence to Nash, however, we demonstrate through extensive experiments the ability of this approach to approximate a unique Nash equilibrium in normal-form games with as many as seven players and twenty one actions (several billion outcomes) that are orders of magnitude larger than those possible with prior algorithms.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

KEYWORDS

Nash; Quantal Response Equilibrium; Limiting Logit Equilibrium; Homotopy; N-player; Normal-form; Empirical Game Theory

ACM Reference Format:
Ian Gemp, Rahul Savani, Marc Lanctot, Yoram Bachrach, Thomas Anthony, Richard Everett, Andrea Tacchetti, Tom Eccles, and János Kramár. 2022. Sample-based Approximation of Nash in Large Many-Player Games via Gradient Descent. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 9 pages.

1 INTRODUCTION

Core concepts from game theory underpin many advances in multiagent systems research. Among these, Nash equilibrium is particularly prevalent. Despite the difficulty of computing a Nash equilibrium [15, 16], a plethora of algorithms [9, 26, 32, 44, 45] and suitable benchmarks [41] have been developed, however, none address large normal-form games with many actions and many players, especially those too big to be stored in memory.

In this work, we develop an algorithm for approximating a Nash equilibrium of a normal-form game with so many actions and players that only a small subset of the possible outcomes in the game can be accessed at a time. We refer the reader to McKelvey and McLennan [34] for a review of approaches for normal-form games. Several algorithms exactly compute a Nash equilibrium for small normal-form games and others efficiently approximate Nash equilibria for special game classes, however, practical algorithms for approximating Nash in large normal-form games with many players, e.g. 7, and many actions, e.g., 21, is lacking. Computational efficiency is of paramount importance for large games because a general normal-form game with n players and m actions contains nm^n payoffs; simply enumerating all payoffs can be intractable and renders classical approaches ineligible. A common approach is to return the profile found by efficient no-regret algorithms that sample payoffs as needed [8] although Flokas et al. [23] recently proved that many from this family do not converge to mixed Nash equilibria in *all* games, 2-player games included.

While significant progress has been made for computing Nash in 2-player normal-form games which can be represented as a *linear* complementarity problem, the many-player setting induces a *nonlinear* complementarity problem, which is "often hopelessly impractical to solve exactly" ([46], p. 105).[1] The combination of high dimensionality (m^n vs m^2 distinct outcomes) and nonlinearity (utilities are degree-n polynomials in the strategies vs degree-2) makes many-player games much more complex.

This more general problem arises in cutting-edge multiagent research when learning [28] and evaluating [1] agents in Diplomacy, a complex 7-player board game. Gray et al. [28] used no-regret learning to approximate a Nash equilibrium of subsampled games, however, this approach is brittle as we show later in Figure 4. In [1], five Diplomacy bots were ranked according to their mass under an approximate Nash equilibrium. We extend that work to encourage convergence to a particular Nash and introduce sampling along with several technical contributions to scale evaluation to 21 Diplomacy bots, a >1000-fold increase in meta-game size.

Equilibrium computation has been an important component of AI in multi-agent systems [46]. It has been (and remains) a critical component of super-human AI in poker [11, 12, 40]. As mentioned above, Nash computation also arises when strategically summarizing a larger domain by learning a lower dimensionality empirical game [51]; such an approach was used in the AlphaStar League, leading to an agent that beat humans in StarCraft [50].Ultimately, this required solving for the Nash of a 2-player, 888-action game, which can take several seconds using state-of-the-art solvers on modern hardware. In contrast, solving an empirical game of Diplomacy, e.g., a 7-player 888-action game, would naively take longer than the current age of the universe. This is well beyond the size of any game we inspect here, however, we approximate the Nash of games several orders of magnitude larger than previously possible, thus taking a step towards this ambitious goal.

Our Contribution: We introduce stochastic optimization into a classical *homotopy* approach resulting in an algorithm that avoids the need to work with the full payoff tensor all at once and is, to our knowledge, the first algorithm generally capable of practically approximating a unique Nash equilibrium in large (billions of outcomes) many-player, many-action normal-form games. We demonstrate our algorithm on 2, 3, 4, 6, 7 and 10 player games (10 in Appx. I; others in §5). We also perform various ablation studies of our algorithm (Appx. F), compare against several baselines including solvers from the popular `Gambit` library (more in Appx. H), and examine a range of domains (more in Appx. I). All appendices can be found in the longer version of this paper [25].

The paper is organized as follows. After formulating the Nash equilibrium problem for a general n-player normal-form game, we review previous work. We discuss how we combine the insights of classical algorithms with ideas from stochastic optimization to develop our final algorithm, *average deviation incentive descent with adaptive sampling*, or ADIDAS. Finally, we compare our proposed algorithm against previous approaches on large games of interest from the literature: games such as Colonel Blotto [2], classical Nash benchmarks from the GAMUT library [41], and games relevant to recent success on the 7-player game Diplomacy [1, 28].

[1]While any n-player game can, in theory, be efficiently solved for approximate equilibria by reducing it to a two-player game, in practice this approach is not feasible for solving large games due to the blowups involved in the reductions. Details in Appx. B.

2 PRELIMINARIES

In a finite n-player game in normal form, each player $i \in \{1, \ldots, n\}$ is given a strategy set $\mathcal{A}_i = \{a_{i1}, \ldots, a_{im_i}\}$ consisting of m_i pure strategies. The pure strategies can be naturally indexed by non-negative integers, so we redefine $\mathcal{A}_i = \{0, \ldots, m_i - 1\}$ as an abuse of notation for convenience. Each player i is also given a payoff or utility function, $u_i : \mathcal{A} \rightarrow \mathbb{R}$ where $\mathcal{A} = \prod_i \mathcal{A}_i$. In games where the cardinality of each player's strategy set is the same, we drop the subscript on m_i. Player i may play a mixed strategy by sampling from a distribution over their pure strategies. Let player i's mixed strategy be represented by a vector $x_i \in \Delta^{m_i-1}$ where Δ^{m_i-1} is the $(m_i - 1)$-dimensional probability simplex embedded in $\mathbb{R}^{m_i}$. Each function u_i is then extended to this domain so that $u_i(\boldsymbol{x}) = \sum_{\boldsymbol{a} \in \mathcal{A}} u_i(\boldsymbol{a}) \prod_j x_{ja_j}$ where $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $a_j \in \mathcal{A}_j$ denotes player j's component of the joint action $\boldsymbol{a} \in \mathcal{A}$. For convenience, let x_{-i} denote all components of $\boldsymbol{x}$ belonging to players other than player i.

We say $\boldsymbol{x} \in \prod_i \Delta^{m_i-1}$ is a Nash equilibrium iff, for all $i \in \{1, \ldots, n\}$, $u_i(z_i, x_{-i}) \leq u_i(\boldsymbol{x})$ for all $z_i \in \Delta^{m_i-1}$, i.e., no player has any incentive to unilaterally deviate from $\boldsymbol{x}$. Nash is most commonly relaxed with ϵ-Nash, an additive approximation: $u_i(z_i, x_{-i}) \leq u_i(\boldsymbol{x}) + \epsilon$ for all $z_i \in \Delta^{m_i-1}$. Later we explore the idea of regularizing utilities with a function S^τ_i (e.g., entropy) as follows:

$$u^\tau_i(\boldsymbol{x}) = u_i(\boldsymbol{x}) + S^\tau_i(x_i, x_{-i}). \tag{1}$$

As an abuse of notation, let the atomic action a_i also denote the m_i-dimensional "one-hot" vector with all zeros aside from a 1 at index a_i; its use should be clear from the context. And for convenience, denote by $H^i_{il} = \mathbb{E}_{x_{-il}}[u_i(a_i, a_l, x_{-il})]$ the Jacobian[2] of player i's utility with respect to x_i and x_l; x_{-il} denotes all strategies belonging to players other than i and l and $u_i(a_i, a_l, x_{-il})$ separates out l's strategy x_l from the rest of the players x_{-i}. We also introduce $\nabla^i_{x_i}$ as player i's utility gradient. Note player i's utility can now be written succinctly as $u_i(x_i, x_{-i}) = x_i^\top \nabla^i_{x_i} = x_i^\top H^i_{il} x_l$ for any l.

In a polymatrix game, interactions between players are limited to local, pairwise games, each of which is represented by matrices H^i_{ij} and H^j_{ij}. This reduces the exponential nm^n payoffs required to represent a general normal form game to a quadratic $n(n-1)m^2$, an efficiency we leverage later.

2.1 Related work

Several approaches exist for computing Nash equilibria of n-player normal form games[3]. Simplicial Subdivision (SD) [49] searches for an equilibrium over discretized simplices; accuracy depends on the grid size which scales exponentially with the number of player actions. Govindan and Wilson [26] propose a homotopy method (GW) that begins with the unique Nash distribution of a game whose payoff tensor has been perturbed by an arbitrary constant tensor.

[2]See Appx. E.2 for an example derivation of the gradient if this form is unfamiliar.

[3]Note that Double-Oracle [37] and PSRO [31] can be extended to n-player games, but require an n-player normal form meta-solver (Nash-solver) and so cannot be considered solvers in their own right. This work provides an approximate meta-solver.

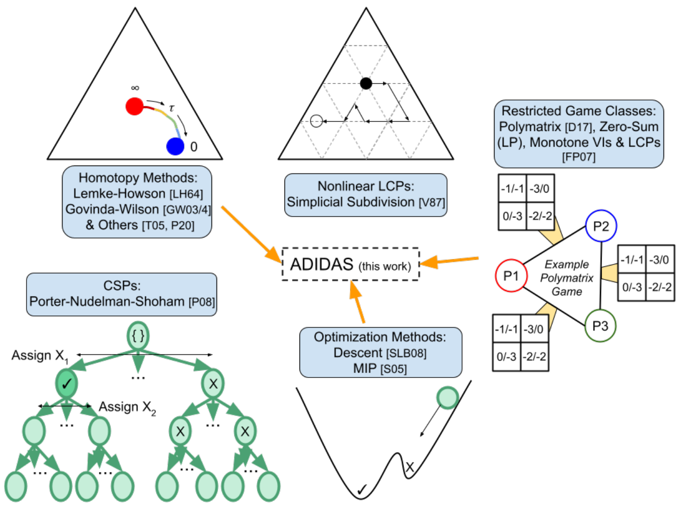


Figure 1: Algorithm Comparison and Overview.

GW then scales back this perturbation while updating the Nash to that of the transformed game. GW is considered an extension of the classic Lemke-Howson algorithm (1964) to 3+ player games (see §4.3, p. 107 of [46]). Another homotopy approach perturbs the payoffs with entropy bonuses, and evolves the Nash distribution along a continuum of quantal response equilibria (QREs) using a predictor-corrector method to integrate a differential equation [48] —we also aim to follow this same continuum. In a slightly different approach, Perolat et al. [43] propose an adaptive regularization scheme that repeatedly solves for the equilibrium of a transformed game. Simple search methods [44] that approach Nash computation as a constraint satisfaction problem appear to scale better than GW and SD as measured on GAMUT benchmarks [41]. Lyapunov approaches minimize non-convex energy functions with the property that zero energy implies Nash [46], however these approaches may suffer from convergence to local minima with positive energy. In some settings, such as polymatrix games with payoffs in $[0,1]$, gradient descent on appropriate energy functions[4] guarantees a $(\frac{1}{2}+\delta)$-Nash in time polynomial in $\frac{1}{\delta}$ [18] and performs well in practice [17].

Teaser. Our proposed algorithm consists of two key conceptual schemes. One lies at the crux of homotopy methods (see Figures 1 and 2). We initialize the Nash approximation, $\boldsymbol{x}$, to the joint uniform distribution, the unique Nash of a game with infinite-temperature entropy regularization. The temperature is then annealed over time. To recover the Nash at each temperature, we minimize an appropriately adapted energy function via (biased) stochastic gradient descent. This minimization approach can be seen as simultaneously learning a suitable polymatrix decomposition of the game similarly to Govindan and Wilson [27] but from batches of stochastic play, i.e., we compute Monte Carlo estimates of the payoffs in the bimatrix game between every pair of players by observing the outcomes of the players' joint actions (sampled from $\boldsymbol{x}$ after each update) rather than computing payoffs as exact expectations.

[4]Equation (2) but with max instead of $\sum$ over player regrets. Note that for symmetric games with symmetric equilibria, these are equivalent up to a multiplicative factor n.

3 DEVIATION INCENTIVE & WARM-UP

We propose minimizing the energy function in equation (2) below, *average deviation incentive* (ADI), to approximate a Nash equilibrium of a large, entropy-regularized normal form game. This loss measures, on average, how much a single agent can exploit the rest of the population by deviating from a joint strategy $\boldsymbol{x}$. For sake of exposition, we drop the normalizing constant from the denominator (number of players, n), and consider the sum instead of the average. This quantity functions as a *loss* that can be minimized over $\mathcal{X} = \prod_i \Delta^{m_i - 1}$ to find a Nash distribution. Note that when ADI is zero, $\boldsymbol{x}$ is a Nash. Also, if $\sum_k$ is replaced by $\max_k$, this loss measures the ϵ of an ϵ-Nash, and therefore, equation (2) is an upper bound on this ϵ. Lastly, note that, in general, this loss function is non-convex and so convergence to local, suboptimal minima is theoretically possible if naively minimizing via first order methods like gradient descent —we explain in §3.1 how we circumvent this pitfall via temperature annealing. Let $\mathsf{BR}_k = \mathsf{BR}(x_{-k}) = \arg\max_{z_k \in \Delta^{m_k-1}} u_k^\tau(z_k, x_{-k})$ be player k's best response to all other players' current strategies where u_k^τ is player k's utility regularized by entropy with temperature τ and formally define

$$\mathcal{L}_{adi}^{\tau}(\boldsymbol{x}) = \sum_k \overbrace{u_k^\tau(\mathsf{BR}_k, x_{-k}) - u_k^\tau(x_k, x_{-k})}^{\text{incentive to deviate to } \mathsf{BR}_k \text{ vs } \boldsymbol{x}_k}. \tag{2}$$

If $\tau = 0$, we drop the superscript and use $\mathcal{L}_{adi}$. The Nash equilibrium of the game regularized with Shannon entropy is called a *quantal response equilibrium*, QRE(τ) (see p. 152-154, 343 of [24]).

Average deviation incentive has been interpreted as a pseudo-distance from Nash in prior work, where it is referred to as `NashConv` [31]. We prefer average deviation incentive because it more precisely describes the function and allows room for exploring alternative losses in future work. The objective can be decomposed into terms that depend on x_k (second term) and x_{-k} (both terms). Minimizing the second term w.r.t. x_k seeks strategies with high utility, while minimizing both terms w.r.t. x_{-k} seeks strategies that cannot be exploited by player k. In reducing $\mathcal{L}_{adi}$, each player k seeks a strategy that not only increases their payoff but also removes others' temptation to exploit them.

A related algorithm is Exploitability Descent (ED) [33]. Rather than minimizing $\mathcal{L}_{adi}$, each player independently maximizes their utility assuming the other players play their best responses. In the two-player normal-form setting, ED is equivalent to extragradient [29] (see Appx. K.2). However, ED is only guaranteed to converge to Nash in two-player, zero-sum games. We include a comparison against ED as well as Fictitious-play, another popular multiagent algorithm, in Appx. H.1. We also relate $\mathcal{L}_{adi}$ to Consensus optimization [39] in Appx. K.1.

3.1 Warm-Up

McKelvey and Palfrey [36] proved the existence of a continuum of QREs starting at the uniform distribution (infinite temperature) and ending at what they called the *limiting logit equilibrium* (LLE).

Furthermore, they showed this path is unique for *almost all games*, partially circumventing the equilibrium selection problem. We encourage the reader to look ahead at Figure 2 for a visual of the homotopy that may prove helpful for the ensuing discussions.

In this work, we assume we are given one of these common games with a unique path (no branching points) so that the LLE is well defined (**Assumption 1**). Furthermore, we assume there exist no "turning points" in the temperature τ along the continuum (**Assumption 2**). Turocy [48] explains that even in generic games, temperature might have to be temporarily increased in order to remain on the path (principal branch) to the LLE. However, Turocy also proves there exists a τ^* such that no turning points exist with $\tau > \tau^*$ suggesting that as long as we remain near the principal branch after τ^*, we can expect to proceed to the LLE.

We follow the principal path by alternating between annealing the temperature and re-solving for the Nash at that temperature by minimizing $\mathcal{L}^{\tau}_{adi}$. We present a basic version of our approach that converges to the limiting logit equilibrium assuming access to exact gradients in Algorithm 1 (proof in Appx. D). We substitute $\lambda = \frac{1}{\tau}$ and initialize $\lambda = 0$ in order to begin at infinite temperature. The proof of this simple warm-up algorithm relies on the detailed examination of the continuum of QREs proposed in [36] and further analyzed in [48]. Theorem 1 presented below is essentially a succinct repetition of one of their known results (Assumptions 3 and 4 below are expanded on in Appx. D). In subsequent sections, we relax the exact gradient assumption and assume gradients are estimated from stochasic play (i.e., each agent samples an action from their side of the current approximation to the Nash).

Algorithm 1 Warm-up: Anneal & Descend

```
Given: Total anneal steps T_λ, total optimizer iterations T*, and
  anneal step size Δλ.
λ = 0
x ← {(1/m_i) 1 ∀ i}
for t_λ = 1 : T_λ do
    λ ← λ + Δλ
    x ← OPT(loss = L_adi^{τ=λ^{-1}}, x_init = x, iters = T*)
end for
return x
```

Theorem 1. *Make assumptions 1 and 2. Also, assume the QREs along the homotopy path have bounded sensitivity to λ given by a parameter σ (Assumption 3), and basins of attraction with radii lower bounded by r (Assumption 4). Let the step size $\Delta\lambda \leq \sigma(r - \epsilon)$ with tolerance ϵ. And let T^* be the supremum over all T such that Assumption 4 is satisfied for any inverse temperature $\lambda \geq \Delta\lambda$. Then, assuming gradient descent for* OPT, *Algorithm 1 converges to the limiting logit equilibrium $\boldsymbol{x}^*_{\lambda=\infty} = \boldsymbol{x}^*_{\tau=0}$ in the limit as $T_\lambda \to \infty$.*

3.2 Evaluating $\mathcal{L}^{\tau}_{adi}$ with Joint Play

In the warm up, we assumed we could compute exact gradients which required access to the entire payoff tensor. However, we want to solve very large games where enumerating the payoff tensor is prohibitively expensive. Therefore, we are particularly interested in minimizing $\mathcal{L}^{\tau}_{adi}$ when only given access to samples of joint play, $\boldsymbol{a} \sim \prod_i x_i$. The best response operator, $\texttt{BR}$, is nonlinear and hence can introduce bias if applied to random samples. For example, consider the game given in Table 1 and assume $x_2 = [0.5, 0.5]^\top$.

u_1	a_{21}	a_{22}
a_{11}	0	0
a_{12}	1	-2
a_{13}	-2	1

u_2	a_{21}	a_{22}
a_{11}	0	0
a_{12}	0	0
a_{13}	0	0

Table 1: A 2-player game with biased stochastic BR's.

Consider computing (row) player 1's best response to a single action sampled from (column) player 2's strategy x_2. Either a_{21} or a_{22} will be sampled with equal probability, which results in a best response of either a_{12} or a_{13} respectively. However, the true expected utilities for each of player 1's actions given player 2's strategy are $[0, -0.5, -0.5]$ for which the best response is the first index, a_{11}. The best response operator completely filters out information on the utility of the true best response a_{11}. Intuitively, a *soft* best response operator, demonstrated in equations (3)-(5), that allows some utility information for each of the actions to pass through could alleviate the problem:

$$\mathbb{E}[\texttt{BR}^{\tau\to 0}] = [0.00, 0.50, 0.50] \quad (3)$$

$$\mathbb{E}[\texttt{BR}^{\tau=1}] \approx [0.26, 0.37, 0.37] \quad (4)$$

$$\mathbb{E}[\texttt{BR}^{\tau=10}] \approx [\mathbf{0.42}, 0.29, 0.29]. \quad (5)$$

By adding an entropy regularizer to the utilities, $\tau\mathcal{H}(x_i)$, we induce a soft-BR. Therefore, the homotopy approach has the added benefit of partially alleviating gradient bias for moderate temperatures. Further empirical analysis of bias can be found in Appx. F.1.

4 ADIDAS

In the previous section, we laid out the conceptual approach we take and identified bias as a potential issue to scaling up computation with Monte Carlo approximation. Here, we inspect the details of our approach, introduce further modifications to reduce the issue of bias, and present our resulting algorithm ADIDAS. Finally, we discuss the advantages of our approach for scaling to large games.

4.1 Deviation Incentive Gradient

Regularizing the utilities with weighted Shannon entropy, $u^{\tau}_k(\boldsymbol{x}) = u_k(\boldsymbol{x}) + S^{\tau}_k(x_k, x_{-k})$, where $S^{\tau}_k(x_k, x_{-k}) = -\tau \sum_{a_k} x_{ka_k} \ln(x_{ka_k})$, leads to the following average deviation incentive gradient derived in Appx. E where $\texttt{BR}_j = \texttt{softmax}(\nabla^j_{x_j}/\tau)$ and $\texttt{diag}(v)$ creates a diagonal matrix with v on the diagonal:

$$\nabla_{x_i}\mathcal{L}^{\tau}_{adi}(\boldsymbol{x}) = -\overbrace{(\nabla^i_{x_i} - \tau(\ln(x_i) + 1))}^{\text{policy gradient}} + \sum_{j\neq i}\Big[J_{x_i}(\texttt{BR}_j)^\top(\nabla^j_{x_j} - \tau(\ln(\texttt{BR}_j) + 1)) + H^j_{ij}(\texttt{BR}_j - x_j)\Big] \quad (6)$$

$$\text{with } J_{x_i}(\texttt{BR}_j) = \frac{1}{\tau}(\texttt{diag}(\texttt{BR}_j) - \texttt{BR}_j\texttt{BR}_j^\top)H^j_{ji}. \quad (7)$$

In the limit, $\nabla_{x_i}\mathcal{L}^{\tau}_{adi}(\boldsymbol{x}) \overset{\tau\to 0^+}{=} -\nabla^i_{x_i} + \sum_{j\neq i} H^j_{ij}(\texttt{BR}_j - x_j)$. The first term is recognized as player i's payoff or *policy* gradient. The second term is a correction that accounts for the other players' incentives to exploit player i through a strategy deviation. Each H^j_{ij} approximates player j's payoffs in the bimatrix game between players i and j. Recall from the preliminaries that in a polymatrix game, these matrices capture the game exactly. We also explore an adaptive Tsallis entropy in Appx. E.

4.2 Amortized Estimates with Historical Play

Section 3.2 discusses the bias that can be introduced when best responding to sampled joint play and how the annealing process of the homotopy method helps alleviate it by softening the BR operator with entropy regularization. To reduce the bias further, we could evaluate more samples from $\boldsymbol{x}$, however, this increases the required computation. Alternatively, assuming strategies have changed minimally over the last few updates (i.e., $\boldsymbol{x}^{(t-2)} \approx \boldsymbol{x}^{(t-1)} \approx \boldsymbol{x}^{(t)}$), we can instead reuse historical play to improve estimates. We accomplish this by introducing an auxiliary variable y_i that computes an exponentially averaged estimate of each player i's payoff gradient $\nabla^i_{x_i}$ throughout the descent similarly to Sutton et al. [47]. We also use y_i to compute an estimate of ADI, $\hat{\mathcal{L}}^{\tau}_{adi}$, as follows:

$$\hat{\mathcal{L}}^{\tau}_{adi}(\boldsymbol{x}, \boldsymbol{y}) = \sum_k y_k^\top(\hat{\mathsf{BR}}_k - x_k) + S^{\tau}_k(\hat{\mathsf{BR}}_k, x_{-k}) - S^{\tau}_k(x_k, x_{-k}) \quad (8)$$

where $\hat{\mathsf{BR}}_k = \arg\max_{z_k \in \Delta^{m_k - 1}} y_k^\top z_k + S^{\tau}_k(z_k, x_{-k})$ is computed with y_k instead of $\nabla^k_{x_k}$. Likewise, replace all $\nabla^k_{x_k}$ with y_k and BR_k with $\hat{\mathsf{BR}}_k$ in equations (6) and (7) when computing the gradient:

4.3 Putting It All Together

Algorithm 2 ADIDAS

1: Given: Strategy learning rate η_x, auxiliary learning rate η_y, initial temperature τ (= 100), ADI threshold ϵ, total iterations T, simulator $\mathcal{G}_i$ that returns player i's payoff given a joint action.
2: $\boldsymbol{x} \leftarrow \{\frac{1}{m_i}\mathbf{1} \;\forall\; i\}$
3: $\boldsymbol{y} \leftarrow \{\mathbf{0} \;\forall\; i\}$
4: **for** $t = 1 : T$ **do**
5:   $a_i \sim x_i \;\forall\; i$
6:   **for** $i \in \{1, \ldots, n\}$ **do**
7:     **for** $j \neq i \in \{1, \ldots, n\}$ **do**
8:       $H^i_{ij}[r, c] \leftarrow \mathcal{G}_i(r, c, a_{-ij}) \;\forall\; r \in \mathcal{A}_i, c \in \mathcal{A}_j$
9:     **end for**
10:   **end for**
11:   $\nabla^i_{x_i} = H^i_{ij} x_j$ for any x_j (or average the result over all j)
12:   $y_i \leftarrow y_i - \max(\frac{1}{t}, \eta_y)(\nabla^i_{x_i} - y_i)$
13:   $x_i \leftarrow x_i - \eta_x \nabla_{x_i} \hat{\mathcal{L}}^{\tau}_{adi}(\boldsymbol{x}, \boldsymbol{y})$ (def. in §4.2 and code in Appx. L)
14:   **if** $\hat{\mathcal{L}}^{\tau}_{adi}(\boldsymbol{x}, \boldsymbol{y}) < \epsilon$ (def. in equation (8)) **then**
15:     $\tau \leftarrow \frac{\tau}{2}$
16:   **end if**
17: **end for**
18: return $\boldsymbol{x}$

Algorithm 2, ADIDAS, is our final algorithm. ADIDAS attempts to approximate the unique continuum of quantal response equilibria by way of a quasi-stationary process—see Figure 2. Whenever the algorithm finds a joint strategy $\boldsymbol{x}$ exhibiting $\hat{\mathcal{L}}^{\tau}_{adi}$ below a threshold ϵ for the game regularized with temperature τ, the temperature is exponentially reduced (line 15 of ADIDAS) as suggested in [48]. Incorporating stochastic optimization into the process enables scaling the classical homotopy approach to extremely large games (large payoff tensors). At the same time, the homotopy approach selects a unique limiting equilibrium and, symbiotically, helps alleviate gradient bias, further amortized by the reuse of historical play.

Limitations: As mentioned earlier, gradient bias precludes a rigorous convergence proof of ADIDAS. However, recent work showed that gradient estimators that are biased, but consistent worked well empirically [14] and follow-up analysis suggests consistency may be an important property [13]. Bias is also being explored in the more complex Riemannian optimization setting where it has been proven that the amount of bias in the gradient shifts the stationary point by a proportional amount [19]. Note that ADIDAS gradients are also consistent in the limit of infinite samples of joint play, and we also find that biased stochastic gradient descent maintains an adequate level of performance for the purpose of our experiments.

No-regret algorithms scale, but have been proven not to converge to Nash [23] and classical solvers [35] converge to Nash, but do not scale. ADIDAS suffers from gradient bias, an issue that may be further mitigated by future research. In this sense, ADIDAS is one of the few, if only, algorithms that can practically approximate Nash in many-player, many-action normal-form games.

Alg Family	Classical	No-Regret	This Work
Convergence to Nash	Yes	No	Yes†
Payoffs Queried	nm^n	Tnm‡	$T(nm)^2$

Table 2: Comparison of solvers. †See *Limitations* in §4.3 and Appx. D.2. ‡Reduce to T at the expense of higher variance.

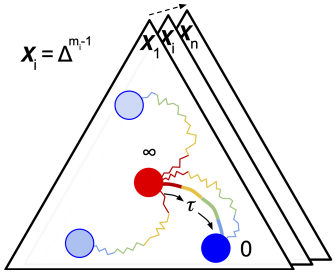
4.4 Complexity and Savings

A normal form game may also be represented with a tensor U in which each entry $U[i, a_1, \ldots, a_n]$ specifies the payoff for player i under the joint action $(a_1, \ldots, a_n)$. In order to demonstrate the computational savings of our approach, we evaluate the ratio of the number of entries in U to the number of entries queried (in the sense of [4, 21, 22]) for computing a single gradient, $\nabla\mathcal{L}^{\tau}_{adi}$. This ratio represents the number of steps that a gradient method can take before it is possible to compute $\mathcal{L}^{\tau}_{adi}$ exactly in expectation.

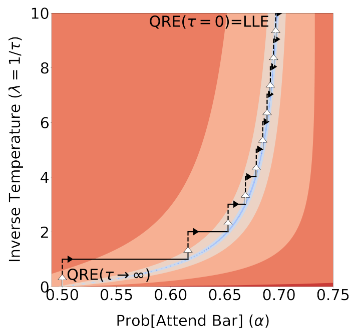
Without further assumptions on the game, the number of entries in a general payoff tensor is nm^n. In contrast, computing the stochastic deviation incentive gradient requires computing H^j_{ij} for all i, j requiring less than $(nm)^2$ entries[5]. The resulting ratio is $\frac{1}{n}m^{n-2}$. For a 7-player, 21-action game, this implies at least 580,000 descent updates can be used by stochastic gradient descent.

If the game is fully symmetric and we desire a symmetric Nash, the payoff tensor can be represented more concisely with $\frac{(m+n-1)!}{n!(m-1)!}$ entries (number of multisets of cardinality n with elements taken

[5]Recall $\nabla^i_{x_i}$ can be computed with $\nabla^i_{x_i} = H^i_{ij} x_j$ for any x_j.



(a) ADIDAS pathologies



(b) 10-player, 2-action El Farol homotopy

Figure 2: (a) In the presence of multiple equilibria, ADIDAS may fail to follow the path to the uniquely defined Nash due to gradient noise, gradient bias, and a coarse annealing schedule. If these issues are severe, they can cause the algorithm to get stuck at a local optimum of $\mathcal{L}^{\tau}_{adi}$—see Figure 5b in §5.2. (b) Such concerns are minimal for the El Farol Bar stage game by Arthur [3]. The solid black curves represent (biased) descent trajectories while the dashed segments indicate the temperature is being annealed.

from a finite set of cardinality m). The number of entries required for a stochastic gradient is less than m^2. Again, for a 7-player 21-action game, this implies at least 2,000 update steps. Although there are fewer unique entries in a symmetric game, we are not aware of libraries that allow sparse storage of or efficient arithmetic on such permutation-invariant tensors. ADIDAS can exploit this symmetry.

5 EXPERIMENTS

We test the performance of ADIDAS empirically on very large games. We begin by considering Colonel Blotto, a deceptively complex challenge domain still under intense research [6, 10], implemented in OpenSpiel [30]. For reference, both the 3 and 4-player variants we consider are an order of magnitude (> 20×) larger than the largest games explored in [44]. We find that no-regret approaches as well as existing methods from Gambit [35] begin to fail at this scale, whereas ADIDAS performs consistently well. At the same time, we empirically validate our design choice regarding amortizing gradient estimates (§4.2). Finally, we end with our most challenging experiment, the approximation of a unique Nash of a 7-player, 21-action (> billion outcome) Diplomacy meta-game.

We use the following notation to indicate variants of the algorithms compared in Table 3. A y superscript prefix, e.g., ^{y}QRE, indicates the estimates of payoff gradients are amortized using historical play; its absence indicates that a fresh estimate is used instead. $\bar{x}_t$ indicates that the average deviation incentive reported is for the average of $\boldsymbol{x}^{(t)}$ over learning. A subscript of ∞ indicates best responses are computed with respect to the true expected payoff gradient (infinite samples). A superscript *auto* indicates the temperature τ is annealed according to line 15 of Algorithm 2. An s in parentheses indicates lines 5-10 of ADIDAS are repeated s times, and the resulting H^i_{ij}'s are averaged for a more accurate estimate. Each game is solved on 1 CPU, except Diplomacy (see Appx. A).

FTRL	Simultaneous Gradient Ascent
RM	Regret-Matching [7]
ATE	ADIDAS with Tsallis (Appx. G)
QRE	ADIDAS with Shannon

Table 3: Algorithms

η_x	$10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$
$\eta_x^{-1} \cdot \eta_y$	1, 10, 100
τ	0.0, 0.01, 0.05, 0.10
$\Pi_\Delta(\nabla \mathcal{L}_{adi})$	Boolean
Bregman-$\psi(\boldsymbol{x})$	$\{\frac{1}{2}\|\|\boldsymbol{x}\|\|^2, -\mathcal{H}(\boldsymbol{x})\}$
ϵ	0.01, 0.05

Table 4: Hyperparameter Sweeps

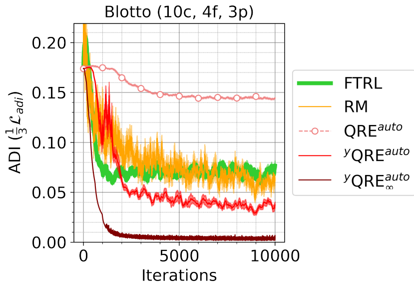
Sweeps are conducted over whether to project gradients onto the simplex ($\Pi_\Delta(\nabla \mathcal{L}_{adi})$), whether to use a Euclidean projection or entropic mirror descent [5] to constrain iterates to the simplex, and over learning rates. Averages over 10 runs of the best hyperparameters are then presented[6] except for Diplomacy for which we present all settings attempted (more in Appx. I.2). Performance is measured by $\mathcal{L}_{adi}$, a.k.a. NashConv [31]. For symmetric games, we enforce searching for a symmetric equilibrium (see Appx. C).

For sake of exposition, we do not present all baselines in all plots, however, we include the full suite of comparisons in the appendix. Our experiments demonstrate that without any additional prior information on the game, ADIDAS is the only practical approach for approximating a Nash equilibrium over many-players and many-actions. We argue this by systematically ruling out other approaches on a range of domains. For example, in Figure 3, RM reduces ADI adequately in Blotto. We do not present RM with improvements in Figure 3 such as using exact expectations, RM_∞, or averaging its iterates, RM($\bar{x}_t$), because we show that both these fail to save RM on the GAMUT game in Figure 4. In other words, we do not present baselines that are unnecessary for logically supporting the claim above. Code is available at github.com/deepmind/open_spiel [30].

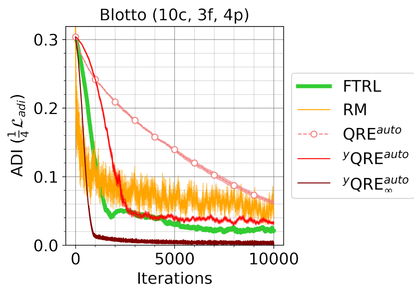
5.1 Med-Scale re. §4.4

Govindan-Wilson is considered a state-of-the-art Nash solver, but it does not scale well to large games. For example, on a symmetric, 4-player Blotto game with 66 actions (10 coins, 3 fields), GW, as

[6]Best hyperparameters are used because we expect running ADIDAS with multiple hyperparameter settings in parallel to be a pragmatic approach to approximating Nash.

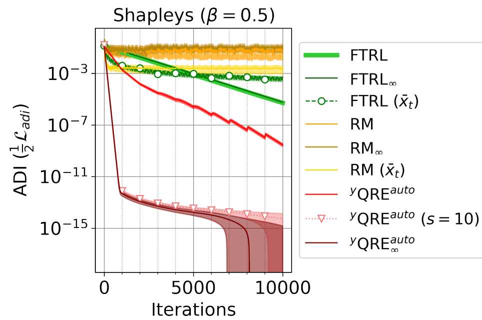


(a) 3-player, 286-action Blotto

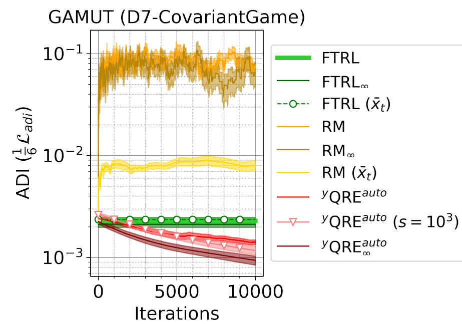


(b) 4-player, 66-action Blotto

Figure 3: Amortizing estimates of joint play using y can reduce gradient bias, further improving performance (e.g., compare QRE^{auto} to ^{y}QRE^{auto} in (a) or (b)).



(a) 2-player, 3-action modified Shapley's



(b) 6-player, 5-action GAMUT-D7

Figure 4: ADIDAS reduces $\mathcal{L}_{adi}$ in both these nonsymmetric games. In contrast, regret matching stalls or diverges in game (a) and diverges in game (b). FTRL makes progress in game (a) but stalls in game (b). In game (a), created by Ostrovski and van Strien [42], to better test performance, x is initialized randomly rather than with the uniform distribution because the Nash is at uniform.

implemented in Gambit, is estimated to take 53,000 hours[7]. Of the solvers implemented in Gambit, none finds a symmetric Nash equilibrium within an hour[8]. Of those, gambit-logit [48] is expected to scale most gracefully. Experiments from the original paper are run on maximum 5-player games (2-actions per player) and 20-action games (2-players), so the 4-player, 66-action game is well outside the original design scope. Attempting to run gambit-logit anyways with a temperature $\tau = 1$ returns an approximate Nash with $\mathcal{L}_{adi} = 0.066$ after 101 minutes. In contrast, Figure 3b shows ADIDAS achieves a lower ADI in ≈ 3 minutes.

Auxiliary y re. §4.2. The introduction of auxiliary variables y_i are supported by the results in Figure 3—^{y}QRE^{auto} significantly improves performance over QRE^{auto} and with low algorithmic cost.

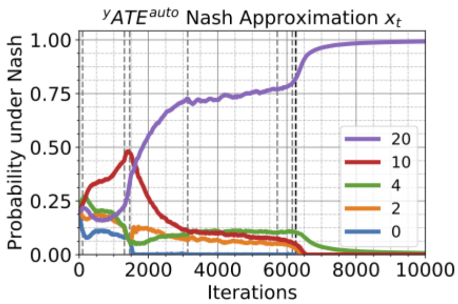
No-regret, No-convergence re. §4.3. In Figure 3, FTRL and RM achieve low ADI quickly in some cases. FTRL has recently been proven not to converge to Nash, and this is suggested to be true of no-regret algorithms in general [23, 38]. Before proceeding, we demonstrate empirically in Figure 4 that FTRL and RM fail on games where ADIDAS significantly reduces ADI. Note that GAMUT (D7) was highlighted as a particularly challenging problem for Nash solvers in [44].

5.2 Large-Scale

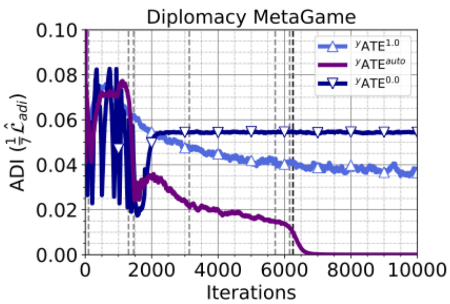
Figure 5 demonstrates an empirical game theoretic analysis [51] of a large symmetric 7-player Diplomacy meta-game where each player elects 1 of 5 trained bots to play on their behalf. Each bot represents a snapshot taken from an RL training run on Diplomacy [1]. In this case, the expected value of each entry in the payoff tensor represents a winrate. Each entry can only be estimated by simulating game play, and the result of each game is a Bernoulli random variable (ruling out deterministic approaches, e.g., gambit). To estimate winrate within 0.01 (ADI within 0.02) of the true estimate with probability 95%, a Chebyshev bound implies more than 223 samples are needed. The symmetric payoff tensor contains 330 unique entries, requiring over 73 thousand games in total. ADIDAS achieves near zero ADI in less than 7 thousand iterations with 50 samples of joint play per iteration ($\approx 5\times$ the size of the tensor).

Continuum of QREs approaching LLE. The purpose of this work is to approximate a unique Nash (the LLE) which ADIDAS is designed to do, however, the approach ADIDAS takes of attempting to track the continuum of QREs (or the continuum defined by the Tsallis entropy) allows returning these intermediate QRE strategies which may be of interest. Access to these intermediate approximations can be useful when a game playing program cannot wait for ADIDAS's final output to play a strategy, for example, in online play. Interestingly, human play appears to track the continuum of QREs in some cases where the human must both learn about the game (rules, payoffs, etc.) whilst also evolving their strategy [36].

[7] Public correspondence with primary gambit developer [link].
[8] gambit-enumpoly returns several non-symmetric, pure Nash equilibria. Solvers listed in Appx. H.2. Symmetric equilibria are necessary for ranking in symmetric meta-games.



(a) 7-player, 5-action symmetric Nash (x_t)



(b) ADI Estimate

Figure 5: (a) Evolution of the symmetric Nash approximation returned by ADIDAS for the 7-player Diplomacy meta-game that considers a subset $\{0, 2, 4, 10, 20\}$ of the available 21 bots; (b) ADI estimated from auxiliary variable y_t. Black vertical lines indicate the temperature τ was annealed.

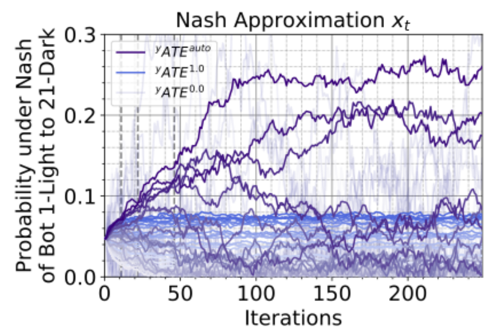
Notice in Figure 5 that the trajectory of the Nash approximation is not monotonic; for example, see the kink around 2000 iterations where bots 10 and 20 swap rank. The continuum of QRE's from $\tau = \infty$ to $\tau = 0$ is known to be complex providing further reason to carefully estimate ADI and its gradients.

Convergence to a Local Optimum. One can also see from Figure 5b that ${}^{y}\text{ATE}^{0.0}$ has converged to a suboptimal local minimum in the energy landscape. This is likely due to the instability and bias in the gradients computed without any entropy bonus; notice the erratic behavior of its ADI within the first 2000 iterations.

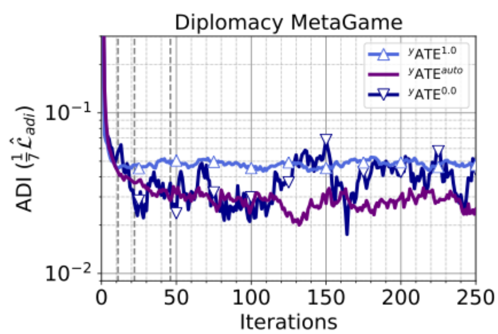
5.3 Very Large-Scale re. §4.4

Finally, we repeat the above analysis with all 21 bots. To estimate winrate within 0.015 (ADI within 0.03) of the true estimate with probability 95%, a Chebyshev bound implies approximately 150 samples are needed. The symmetric payoff tensor contains 888, 030 unique entries, requiring over 100 million games in total. Note that ignoring the symmetry would require simulating $150 \times 21^7 \approx 270$ billion games and computing over a trillion payoffs (×7 players). Simulating all games, as we show, is unnecessarily wasteful, and just storing the entire payoff tensor in memory, let alone computing with it would be prohibitive without special permutation-invariant data structures (≈ 50GB with `float32`). In Figure 6a, ADIDAS with $\eta_x = \eta_y = 0.1$ and $\epsilon = 0.001$ achieves a stable ADI below 0.03 in less than 100 iterations with 10 samples of joint play per iteration and each game repeated 7 times (< 2.5% of the games run by the naive alternative). As expected, bots later in training (darker lines) have higher mass under the Nash distribution computed by ${}^{y}\text{ATE}^{auto}$. Runtime is discussed in Appx. A.

Importance of Entropy Bonus. Figure 6a shows how the automated annealing mechanism (${}^{y}\text{ATE}^{auto}$) seeks to maintain entropy regularization near a "sweet spot" —too little entropy (${}^{y}\text{ATE}^{0.0}$) results in an erratic evolution of the Nash approximation and too much entropy (${}^{y}\text{ATE}^{1.0}$) prevents significant movement from the initial uniform distribution. Figure 6b shows that ADIDAS with the automated annealing mechanism meant to trace the QRE continuum achieves a lower ADI than its fixed temperature variants.



(a) 7-player, 21-action symmetric Nash (x_t)



(b) ADI Estimate

Figure 6: (a) Evolution of the symmetric Nash approximation returned by ADIDAS for the 7-player, 21-bot Diplomacy meta-game; (b) ADI estimated from auxiliary variable y_t. Black vertical lines indicate the temperature τ was annealed.

In the appendix, we perform additional ablation studies (e.g., no entropy, annealing), measure accuracy of $\hat{\mathcal{L}}^{\tau}_{adi}$, compare against more algorithms on other domains, and consider Tsallis entropy.

6 CONCLUSION

Existing algorithms either converge to Nash, but do not scale to large games or scale to large games, but do not converge to Nash. We proposed an algorithm to fill this void that queries necessary payoffs through sampling, obviating storing the full payoff tensor in memory. ADIDAS is principled and shown empirically to approximate Nash in large-normal form games.

REFERENCES

[1] Thomas Anthony, Tom Eccles, Andrea Tacchetti, János Kramár, Ian Gemp, Thomas C Hudson, Nicolas Porcel, Marc Lanctot, Julien Pérolat, Richard Everett, et al. 2020. Learning to Play No-Press Diplomacy with Best Response Policy Iteration. In *Advances in Neural Information Processing Systems*.

[2] Ayala Arad and Ariel Rubinstein. 2012. Multi-dimensional iterative reasoning in action: The case of the Colonel Blotto game. *Journal of Economic Behavior & Organization* 84, 2 (2012), 571–585.

[3] W Brian Arthur. 1994. Complexity in economic theory: Inductive reasoning and bounded rationality. *The American Economic Review* 84, 2 (1994), 406–411.

[4] Yakov Babichenko. 2016. Query complexity of approximate Nash equilibria. *Journal of the ACM (JACM)* 63, 4 (2016), 36:1–36:24.

[5] Amir Beck and Marc Teboulle. 2003. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters* 31, 3 (2003), 167–175.

[6] Soheil Behnezhad, Sina Dehghani, Mahsa Derakhshan, MohammadTaghi Haji-Aghayi, and Saeed Seddighin. 2017. Faster and simpler algorithm for optimal strategies of Blotto game. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 369–375.

[7] David Blackwell et al. 1956. An analog of the minimax theorem for vector payoffs. *Pacific J. Math.* 6, 1 (1956), 1–8.

[8] Avrim Blum and Yishay Mansour. 2007. *Learning, Regret Minimization, and Equilibria*. Cambridge University Press, 79–102. https://doi.org/10.1017/CBO9780511800481.006

[9] Ben Blum, Christian R Shelton, and Daphne Koller. 2006. A continuation method for Nash equilibria in structured games. *Journal of Artificial Intelligence Research* 25 (2006), 457–502.

[10] Enric Boix-Adserà, Benjamin L Edelman, and Siddhartha Jayanti. 2020. The Multiplayer Colonel Blotto Game. In *Proceedings of the 21st ACM Conference on Economics and Computation*. 47–48.

[11] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. 2015. Heads-up Limit Hold'em Poker is Solved. *Science* 347, 6218 (January 2015), 145–149.

[12] Noam Brown and Tuomas Sandholm. 2017. Superhuman AI for Heads-up No-limit Poker: Libratus beats top professionals. *Science* 360, 6385 (December 2017).

[13] Jie Chen and Ronny Luss. 2018. Stochastic gradient descent with biased but consistent gradient estimators. *arXiv preprint arXiv:1807.11880* (2018).

[14] Jie Chen, Tengfei Ma, and Cao Xiao. 2018. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247* (2018).

[15] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. 2009. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM (JACM)* 56, 3 (2009), 1–57.

[16] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. 2009. The complexity of computing a Nash equilibrium. *SIAM J. Comput.* 39, 1 (2009), 195–259.

[17] Argyrios Deligkas, John Fearnley, Tobenna Peter Igwe, and Rahul Savani. 2016. An empirical study on computing equilibria in polymatrix games. *arXiv preprint arXiv:1602.06865* (2016).

[18] Argyrios Deligkas, John Fearnley, Rahul Savani, and Paul Spirakis. 2017. Computing approximate Nash equilibria in polymatrix games. *Algorithmica* 77, 2 (2017), 487–514.

[19] Alain Durmus, Pablo Jiménez, Éric Moulines, Salem Said, and Hoi-To Wai. 2020. Convergence analysis of Riemannian stochastic approximation schemes. *arXiv preprint arXiv:2005.13284* (2020).

[20] Francisco Facchinei and Jong-Shi Pang. 2007. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media.

[21] John Fearnley, Martin Gairing, Paul W Goldberg, and Rahul Savani. 2015. Learning equilibria of games via payoff queries. *The Journal of Machine Learning Research* 16, 1 (2015), 1305–1344.

[22] John Fearnley and Rahul Savani. 2016. Finding approximate Nash equilibria of bimatrix games via payoff queries. *ACM Transactions on Economics and Computation (TEAC)* 4, 4 (2016), 25:1–25:19.

[23] Lampros Flokas, Emmanouil-Vasileios Vlatakis-Gkaragkounis, Thanasis Lianeas, Panayotis Mertikopoulos, and Georgios Piliouras. 2020. No-regret learning and mixed Nash equilibria: They do not mix. *arXiv preprint arXiv:2010.09514* (2020).

[24] Drew Fudenberg, David K Levine, et al. 1998. The Theory of Learning in Games. *MIT Press Books* 1 (1998).

[25] Ian Gemp, Rahul Savani, Marc Lanctot, Yoram Bachrach, Thomas Anthony, Richard Everett, Andrea Tacchetti, Tom Eccles, and János Kramár. 2022. Sample-based Approximation of Nash in Large Many-Player Games via Gradient Descent. *arXiv preprint arXiv:2106.01285* (2022).

[26] Srihari Govindan and Robert Wilson. 2003. A global Newton method to compute Nash equilibria. *Journal of Economic Theory* 110, 1 (2003), 65–86.

[27] Srihari Govindan and Robert Wilson. 2004. Computing Nash equilibria by iterated polymatrix approximation. *Journal of Economic Dynamics and Control* 28, 7 (2004), 1229–1241.

[28] Jonathan Gray, Adam Lerer, Anton Bakhtin, and Noam Brown. 2020. Human-Level Performance in No-Press Diplomacy via Equilibrium Search. *arXiv preprint arXiv:2010.02923* (2020).

[29] Galina M Korpelevich. 1976. The extragradient method for finding saddle points and other problems. *Matecon* 12 (1976), 747–756.

[30] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. 2019. OpenSpiel: A Framework for Reinforcement Learning in Games. *CoRR* abs/1908.09453 (2019). arXiv:1908.09453 [cs.LG] http://arxiv.org/abs/1908.09453

[31] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*. 4190–4203.

[32] Carlton Lemke and Joseph Howson, Jr. 1964. Equilibrium points of bimatrix games. *Journal of the Society for industrial and Applied Mathematics* 12, 2 (1964), 413–423.

[33] Edward Lockhart, Marc Lanctot, Julien Pérolat, Jean-Baptiste Lespiau, Dustin Morrill, Finbarr Timbers, and Karl Tuyls. 2019. Computing Approximate Equilibria in Sequential Adversarial Games by Exploitability Descent. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*.

[34] Richard D McKelvey and Andrew McLennan. 1996. Computation of equilibria in finite games. *Handbook of Computational Economics* 1 (1996), 87–142.

[35] Richard D McKelvey, Andrew M McLennan, and Theodore L Turocy. 2016. Gambit: Software tools for game theory, version 16.0.1.

[36] Richard D McKelvey and Thomas R Palfrey. 1995. Quantal response equilibria for normal form games. *Games and Economic Behavior* 10, 1 (1995), 6–38.

[37] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. 2003. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 536–543.

[38] Panayotis Mertikopoulos, Christos Papadimitriou, and Georgios Piliouras. 2018. Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2703–2717.

[39] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. 2017. The Numerics of GANs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 1823–1833.

[40] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. 2017. DeepStack: Expert-level artificial intelligence in Heads-up No-limit Poker. *Science* 358, 6362 (October 2017).

[41] Eugene Nudelman, Jennifer Wortman, Yoav Shoham, and Kevin Leyton-Brown. 2004. Run the Gamut: A comprehensive approach to evaluating game-theoretic algorithms. In *AAMAS*, Vol. 4. 880–887.

[42] Georg Ostrovski and Sebastian van Strien. 2013. Payoff performance of fictitious play. *arXiv preprint arXiv:1308.4049* (2013).

[43] Julien Perolat, Remi Munos, Jean-Baptiste Lespiau, Shayegan Omidshafiei, Mark Rowland, Pedro Ortega, Neil Burch, Thomas Anthony, David Balduzzi, Bart De Vylder, et al. 2020. From Poincaré Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization. *arXiv preprint arXiv:2002.08456* (2020).

[44] Ryan Porter, Eugene Nudelman, and Yoav Shoham. 2008. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior* 63, 2 (2008), 642–662.

[45] Tuomas Sandholm, Andrew Gilpin, and Vincent Conitzer. 2005. Mixed-integer programming methods for finding Nash equilibria. In *AAAI*. 495–501.

[46] Yoav Shoham and Kevin Leyton-Brown. 2009. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

[47] Richard S Sutton, Csaba Szepesvári, and Hamid Reza Maei. 2008. A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation. *Advances in Neural Information Processing Systems* 21, 21 (2008), 1609–1616.

[48] Theodore L Turocy. 2005. A dynamic homotopy interpretation of the logistic quantal response equilibrium correspondence. *Games and Economic Behavior* 51, 2 (2005), 243–263.

[49] Gerard van der Laan, AJJ Talman, and L Van der Heyden. 1987. Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling. *Mathematics of Operations Research* 12, 3 (1987), 377–397.

[50] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[51] Michael P Wellman. 2006. Methods for empirical game-theoretic analysis. In *AAAI*. 1552–1556.