Solving infinite games with deep multiagent reinforcement learning

Thesis proposal

Carlos G. Martin
Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis committee
Tuomas Sandholm (CMU, chair)
Vincent Conitzer (CMU)
Fei Fang (CMU)
Ian Gemp (Google)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

April 2026

Abstract

Game theory provides a rigorous mathematical framework for analyzing strategic interactions between agents with interdependent goals. It can be used to prescribe rational behavior for individuals, predict emergent dynamics in multiagent systems, and design effective social mechanisms.

In this thesis, we study the problem of solving infinite games. Such games can have infinitely many states, actions, players, and steps. Unlike mean-field games, our players need not be symmetric or exchangeable. Furthermore, we allow such games to have partial observability, hidden information, imperfect recall, stochastic state transitions, discontinuous utility functions, and interdependent social preferences (i.e., matrix-valued discount factors). Together, these properties can model a wide range of highly complex, real-world scenarios that defy traditional game-theoretic solvers.

To tackle this problem, we propose a unified framework grounded in deep multiagent reinforcement learning. It includes five core components. First, it introduces randomized policy networks (RPNs) to model observation-dependent mixed strategies over infinite action spaces. Second, it represents complex strategy profiles across an infinite continuum of players using player-to-strategy networks (P2SNs). Third, it evolves these representations through a shared-parameter simultaneous gradient (SPSG), which extends the standard simultaneous gradient to this shared-parameter regime. Fourth, to ensure computational efficiency, it estimates this gradient using randomized parameter perturbations via a joint-perturbation simultaneous pseudo-gradient (JPSPG). Fifth, it employs approximate exploitability descent (ApproxED) with learned best-response functions (BRFs).

We propose to benchmark our approach on a diverse suite of real-world domains. These include financial markets, traffic flow, epidemiological contagion, energy grids, and population ecology.

Acknowledgments

This material is based on work supported by the following sources of funding:

- **National Science Foundation** (NSF) grants
  - IIS-1901403
  - CCF-1733556
  - RI-2312342
  - RI-1901403
- **Army Research Office** (ARO) awards
  - W911NF2210266
- **Vannevar Bush Faculty Fellowship** (VBFF) **Office of Naval Research** (ONR) awards
  - N00014-23-1-2876
- **National Institutes of Health** (NIH) awards
  - A240108S001

Contents

Chapter 1

Introduction

1.1 Background

Game theory is the study of strategic interactions between agents. Roger Myerson, in *Game Theory: Analysis of Conflict* (Myerson, 1991), defines it as "the study of mathematical models of conflict and cooperation between intelligent rational decision-makers." In these models, agents may have independent interests, creating incentives for pure cooperation, pure competition, or a complicated mixture of the two.

Within this framework, a **game** is a setting in which such agents interact. Each player has a set of **strategies** to choose from. A **strategy profile** is a complete assignment of strategies to players from their strategy sets. Finally, a **utility function** defines the total reward attained by each player, given all players' strategies.

The central solution concept in game theory is the **Nash equilibrium** (NE), named after mathematician John Nash. It is a strategy profile for which no player has an incentive to unilaterally change their strategy. In his foundational 1951 paper, *Non-Cooperative Games* (Nash, 1951), Nash formally defined it as follows: "An equilibrium point is an n-tuple such that each player's mixed strategy maximizes his payoff if the strategies of the others are held fixed. Thus each player's strategy is optimal against those of the others." Martin Osborne, in *An Introduction to Game Theory* (Osborne, 2004), defines a Nash equilibrium as a strategy profile "with the property that no player can do better by changing his strategy, given the other players' strategies."

Machine learning (ML) is a subset of artificial intelligence focused on building systems that learn from data. In his foundational textbook *Machine Learning* (Mitchell, 1997), Tom Mitchell provided the most widely accepted formal definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." Most real-world problems have a learnable structure, which is why inductive reasoning and ML work well (Goldblum et al., 2024).

Neural networks (NNs) are computational models inspired by biological neural architectures and serve as a foundational pillar of modern ML. McCulloch and Pitts (1943) introduced the concept of artificial neurons. Rosenblatt (1958) introduced the perceptron, an architecture and algorithm for supervised learning of binary classifiers. A defining characteristic of neural networks is their capacity to act as universal function approximators while maintaining a robust ability to generalize across unseen inputs.

Specifically, the classical **universal approximation theorem** (UAT) (Cybenko, 1989; Hornik, 1991; Leshno et al., 1993; Pinkus, 1999) establishes that a feedforward network with a single, sufficiently wide hidden layer can approximate arbitrary continuous functions on a compact domain. While this guarantees the theoretical representational power of shallow networks, modern architectures typically rely on **depth** (multiple hidden layers) rather than extreme width to achieve highly efficient, hierarchical feature extraction.

Deep learning (DL) allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction (LeCun, Bengio, and Hinton, 2015). Subsequent extensions of the UAT specifically address DL, mathematically demonstrating that depth can exponentially reduce the required number of neurons for certain classes of functions (Telgarsky, 2016; Lin, Tegmark, and Rolnick, 2017). Furthermore, modern variants of the theorem establish that deep networks with bounded width (often strictly related to the input dimension) but arbitrary depth are also universal approximators (Lu et al., 2017; Yarotsky, 2017; Kidger and Lyons, 2020).

Reinforcement learning (RL) is a type of ML in which autonomous agents learn to make decisions by interacting with their environment. Richard Sutton and Andrew Barto, in their seminal work *Reinforcement Learning: An Introduction* (Sutton and Barto, 1998), defined RL as "learning what to do—how to map situations to actions—so as to maximize a numerical reward signal." By receiving rewards for desired behaviors and penalties for undesirable ones, the agent discovers how to maximize cumulative rewards over time. It is widely used for complex, sequential tasks like game-playing, robotics, and personalized recommendations.

Multiagent reinforcement learning (MARL) sits at the intersection of game theory and ML. As described in *A Comprehensive Survey of Multiagent Reinforcement Learning* (Busoniu, Babuska, and Schutter, 2008), MARL investigates systems where "multiple agents learn by dynamically interacting with their environment and with each other." Because the environment is made non-stationary by the concurrent learning of other agents, MARL algorithms must incorporate the strategic, game-theoretic reasoning required to anticipate and respond to other intelligent actors.

Real-world agents—humans and machines alike—are neither omniscient nor computationally unbounded. However, they are typically expected to act *approximately* rationally within their limitations. This idea, introduced by Herbert Simon in Simon (1955) and termed **bounded rationality**, holds that agents optimize subject to constraints on information, cognitive capacity, and time. This observation motivates solution concepts like ε-Nash equilibrium (Definition 85), in which each player's regret is at most ε rather than exactly zero.

Despite human cognitive limitations, the rigorous mathematical prescriptions of game theory have proven highly effective when deployed by artificial agents against actual humans. For example, *DeepStack* (Moravčík et al., 2017) and *Libratus* (Brown and Sandholm, 2018) were the first AIs to achieve superhuman performance in heads-up (2-player) no-limit Texas Hold'em. Similarly, *Pluribus* (Brown and Sandholm, 2019) was the first AI to achieve superhuman performance in multi-player no-limit Texas Hold'em. By mastering the world's most popular poker variant, these AIs demonstrated that game-theoretic algorithms can successfully navigate hidden information, deception, and human bounded rationality on a massive scale.

1.2 Goal

The main goal of this project is to develop a **unified algorithmic framework** capable of computing approximate Nash equilibria in highly complex, realistic game-theoretic environments. Specifically,

our goal is to design a framework that robustly handles games that have the following **complexities**:

- **Complex information structures:**
  - Partial observability of the game state.
  - Hidden information (players know things others do not).
  - Imperfect recall (players might forget previously known information).
- **Infinite scale and dimensionality:**
  - Infinitely many states.
  - Infinitely many actions (and the computation of mixed strategies over them).
  - Infinitely many players (in a general setting strictly beyond standard mean-field games, where players are not necessarily symmetric or indistinguishable and may possess unique strategy sets and payoffs).
  - Infinitely many steps (e.g., differential games).
- **Intricate dynamics and payoffs:**
  - Multiple steps (i.e., dynamic games).
  - Stochastic observations, rewards, and state transitions.
  - Discontinuous utility functions (e.g., to accurately model auctions).
  - Matrix-valued discount factors (representing agents valuing each other's future rewards, naturally capturing social dynamics like altruism and spite).

Standard equilibrium-finding techniques fail to scale or even apply to this highly-complex setting. We develop our framework to address all of these complexities simultaneously. We apply it to a suite of **simulated environments** that mirror real-world applications. We evaluate it using standard game-theoretic performance metrics. In particular, we estimate the empirical **exploitability** to quantify the robustness of the computed strategies. To tackle the intractability of exact exploitability computation in these domains, we estimate empirical exploitability as follows:

- Conditional sampling and action space discretization for single-step Bayesian games.
- Training best-response agents via reinforcement learning for multi-step dynamic games.

1.3 Motivation

In the next few subsections, we provide theoretical and practical motivations for some of the key aspects of the very general setting we are tackling.

1.3.1 Infinite-action games

While most research on computing game-theoretic equilibria focuses on finite, discrete action spaces, many real-world settings inherently involve continuous domains—such as allocations of **space, time, or money**. Notable examples include continuous resource allocation games (Ganzfried, 2021), security games in continuous spaces (Kamra, Fang, et al., 2017; Kamra, Gupta, Fang, et al., 2018;

Kamra, Gupta, Wang, et al., 2019), network games (Ghosh and Kundu, 2019), military simulations and wargames (Marchesi, Trovò, and Gatti, 2020), and complex video games (Berner et al., 2019; Vinyals et al., 2019). Furthermore, even when an action space is technically discrete, it can be so fine-grained that treating it as continuous yields significant computational efficiency (Borel and Ville, 1938; Chen and Ankenman, 2006; Ganzfried and Sandholm, 2010).

These infinite-action games often introduce additional complexities. For example, **discontinuous utility functions** are highly relevant to real-world scenarios, particularly in economic settings like auctions. Finally, because players must often randomize their actions to avoid being exploited (assuming access to randomization oracles), it is essential to be able to represent and learn **mixed strategies** over these infinite action spaces.

1.3.2 Infinite-player games

Many disciplines—ranging from economics and financial markets to congestion analysis, crowd dynamics, epidemiology, and population ecology—study systems comprising **large numbers of interacting agents**. The presence of numerous entities, each pursuing individual and often competing interests, makes modeling and analyzing these multiagent systems notoriously complex. To simplify this analysis, researchers frequently model such systems as *infinite*-player games. Taking the limit as the number of players approaches infinity yields a mathematical tractability akin to the treatment of **many-particle systems** in thermodynamics and statistical mechanics.

This continuous approximation has a rich history. A foundational paper by Aumann (1964) proposed modeling markets with individually insignificant traders as a continuum, analogous to the continuous points on a line. Building on this concept, **mean-field game** (MFG) theory studies the limit of symmetric N-player games as N approaches infinity (Lasry and Lions, 2007; Huang, Malhamé, and Caines, 2006; Huang, Caines, and Malhamé, 2007). In an MFG, agents are individually negligible and exchangeable. Rather than tracking every pairwise interaction, each infinitesimal agent optimizes its control against a **mean field** representing the aggregate state distribution of the population. Equilibrium is thus defined as a self-consistent mean field generated by the agents' best responses.

Computing Nash equilibria of these many-agent systems is invaluable not only for an individual agent determining its optimal strategy, but also for social scientists seeking to understand the system from the outside, and for social planners aiming to direct it. In these contexts, the **aggregate behavior** of players is often what truly matters, rather than the specific actions of individuals. In mechanism design, for instance, social planners strive to engineer games whose equilibria possess desired properties.

Consequently, this is highly applicable to real-world settings such as many-player auctions, financial markets, and transit networks. From a theoretical perspective, it can also be utilized to study the equilibria of environments like **abstract economies**. Ultimately, this framework can be employed both to make precise prescriptions for how individual agents should act rationally, *and* to predict the emergent behavior of self-interested agents within a massive system.

1.3.3 Matrix-valued discount factors

A **discount factor** quantifies how much an agent values future rewards relative to immediate ones. In a multiagent game, each player can have a distinct discount factor. Traditionally, these are represented as a *discount vector*, where each component corresponds to an individual player.

This concept can be generalized to *matrix-valued* discount factors. The diagonal elements of such a matrix represent the standard, individual discount factors, while the off-diagonal elements capture cross-player interactions. A non-zero off-diagonal factor indicates that an agent internalizes *another* agent's future rewards, either positively or negatively.

Assuming a parameter $\alpha > 0$, we can model a variety of distinct behavioral dynamics between two agents using the following 2×2 matrices.

- **Egoism (strict self-interest)**: Both agents are strictly self-interested and perfectly indifferent to their counterpart's outcomes.

$$\gamma_{\text{egoists}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{1.1}$$

- **Lovers (mutual altruism)**: Both agents value their counterpart's happiness in addition to their own.

$$\gamma_{\text{lovers}} = \begin{bmatrix} 1 & \alpha \\ \alpha & 1 \end{bmatrix} \tag{1.2}$$

- **Bitter enemies (mutual spite)**: Both agents actively derive disutility from their counterpart's success.

$$\gamma_{\text{enemies}} = \begin{bmatrix} 1 & -\alpha \\ -\alpha & 1 \end{bmatrix} \tag{1.3}$$

- **The unrequited lover (one-sided altruism)**: Player 1 values Player 2's happiness, but Player 2 is entirely indifferent to Player 1.

$$\gamma_{\text{unrequited}} = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \tag{1.4}$$

- **The saboteur (one-sided spite)**: Player 1 actively dislikes Player 2 and seeks to minimize their counterpart's reward, while Player 2 remains a pure egoist.

$$\gamma_{\text{saboteur}} = \begin{bmatrix} 1 & -\alpha \\ 0 & 1 \end{bmatrix} \tag{1.5}$$

- **Hyper-competitive (zero-sum mindset)**: The agents care only about the *relative difference* in their rewards. A counterpart's gain is mathematically identical to a personal loss.

$$\gamma_{\text{zero-sum}} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{1.6}$$

- **Martyrs (pure self-sacrifice, self-abnegating altruism)**: Both agents place zero value on their own rewards and care *only* about the outcomes of their counterpart.

$$\gamma_{\text{martyrs}} = \begin{bmatrix} 0 & \alpha \\ \alpha & 0 \end{bmatrix} \tag{1.7}$$

In game theory and sociobiology, these interactive preferences are commonly studied under the name of **altruism and spite**. There is a rich body of literature that explores how such motives cause behavior to deviate from strict self-interest.

In behavioral and experimental economics, this class of preferences is known as other-regarding or **social preferences**. Fehr and Schmidt (1999) proposed the canonical model of **inequity aversion**, in which an agent's utility is penalized by both advantageous and disadvantageous deviations from their counterpart's payoff. Bolton and Ockenfels (2000) proposed the related ERC (equity, reciprocity, and competition) framework, formulated in terms of relative rather than absolute comparisons. Charness and Rabin (2002) tested these models experimentally, disentangling inequity aversion from broader concerns for social welfare and reciprocity. Collectively, these models motivate the explicit parameterization of interpersonal weightings in utility. This is precisely the role played by the off-diagonal entries of our matrix-valued discount factors.

For instance, in the context of public goods, Saijo and Nakamura (1995) examined deviations from strict economic rationality—where pure self-interest dictates either full contribution or free-riding—highlighting the role of spite in voluntary contribution mechanisms. Levine (1998) proposed a broader theory of altruism where a player's payoff is a linear combination of their own and their counterpart's monetary income. By analyzing ultimatum and centipede game experiments, they estimated the population distribution of altruism and spite, noting that the weight placed on a counterpart's income relies on private information and beliefs about the counterpart's disposition.

The impact of spite is particularly well-documented in **auction theory**, where agents frequently bid more aggressively than standard models prescribe. This aggressiveness can stem from inherent psychological spite or strategic advantages, such as long-term benefits in closed markets where a competitor's loss translates to personal gain. Morgan, Steiglitz, and Reis (2003) demonstrated that bidders who attach disutility to rival surplus bid aggressively enough to rationalize anomalies observed in laboratory settings. Brandt, Sandholm, and Shoham (2007) corroborated this by analyzing spiteful behavior in sealed-bid auctions. Expanding on these foundations, Sharma and Sandholm (2010) introduced the first analysis of asymmetric spite, reflecting the realistic scenario where bidders are spiteful to varying degrees. Finally, addressing the ultimate mechanism design challenge in this domain, Tang and Sandholm (2012) derived optimal auctions tailored specifically for both spiteful and altruistic bidders.

Beyond static or single-step interactions, the concept of generalized discounting has been extended to sequential decision-making and dynamic environments. In the context of temporal discounting, White (2017) introduced a reinforcement learning task formalism that generalizes the traditional scalar discount factor into a **transition-dependent variable**. Building on this flexibility, Pitis (2019) provided an axiomatic, decision-theoretic proof demonstrating that standard, fixed scalar discounting is fundamentally insufficient for modeling general human preferences. This expanded framework accommodates a state-action-dependent discount factor that **can be greater than 1**, provided there is eventual long-run discounting.

Parallel advancements have formalized social discounting in multi-step settings, where the matrix values represent **interpersonal weights** rather than temporal horizons. McKee et al. (2020) integrated **social value orientation** (SVO) directly into deep multiagent reinforcement learning. They demonstrated that introducing heterogeneity in SVOs generates meaningful and complex behavioral variation among agents, closely mirroring the predictions of interdependence theory. Formalizing the mathematical limits of these interactions, Ray and Vohra (2020) studied strategic environments characterized by **payoff-based externalities**—situations where an agent's utility depends not only on their own actions but directly on the realized payoffs of others. They termed

the resulting systems "**games of love and hate**", providing a rigorous foundation for the infinitely reflecting utilities that arise within these interdependent matrices.

1.4 Outline

The structure of this thesis proposal is as follows. In Section 2, we define the mathematical notation and concepts that are required for our setting, which deals with infinite spaces. In Section 3, we describe the game-theoretic concepts that are needed for our setting, and give a mathematically rigorous formulation of the problem we are tackling. In Section 4, we describe prior work that has been done in each of the main areas we are tackling. In Section 5, we describe the work we have already completed that contributes toward our ultimate goal. In Section 6, we describe the next steps for our project and proposed future work. In the appendix, we include additional material. This includes additional experiments and theoretical analysis. Section A describes other works I completed during my PhD that fall outside the scope of this thesis.

Chapter 2

Mathematical preliminaries

In this section, we define the mathematical notation and concepts that are required for our setting. Since we deal with infinite spaces, we need to define related concepts rigorously.

2.1 Notation

In the expressions below, A and B are sets, f and g are functions, n is a natural number, ϕ is a formula, X is a random variable, d is a metric, and V is a Banach space.

$x \in A$	x is an element of A	
$A \subseteq B$	A is a subset of B	$\forall x \in A, x \in B$
$A \cup B$	union of A and B	$\{x \mid x \in A \vee x \in B\}$
$A \cap B$	intersection of A and B	$\{x \mid x \in A \wedge x \in B\}$
$A \setminus B$	relative complement of B in A	$\{x \mid x \in A \wedge x \notin B\}$
$\mathcal{P}(A)$	powerset of A	$\{B \mid B \subseteq A\}$
$\cup A$	union of A	$\{x \mid \exists B \in A, x \in B\}$
$\varnothing$	empty set	$\{x \mid \bot\}$
$\sup A$	supremum of A	
$\inf A$	infimum of A	
$\operatorname{dom} f$	domain of f	
$\operatorname{cod} f$	codomain of f	
$A \times B$	Cartesian product of A and B	
$A \to B$ or B^A	set of functions from A to B	
$f[A]$	image of A under f	$\{f(x) \mid x \in A\}$
$f^{-1}[A]$	preimage of A under f	$\{x \mid f(x) \in A\}$
$\operatorname{im} f$	image of f	$f[\operatorname{dom} f]$
$\operatorname{zero} f$	zeroset of f	$f^{-1}[\{0\}]$
$\operatorname{supp} f$	support of f	$(\operatorname{dom} f) \setminus (\operatorname{zero} f)$
$\operatorname{argmax} f$	argmax of f	$f^{-1}[\{\sup(\operatorname{im} f)\}]$
$\operatorname{argmin} f$	argmin of f	$f^{-1}[\{\inf(\operatorname{im} f)\}]$
Πf	Cartesian product of f	
$f \circ g$	composition of f and g	

$\mathbb{N}$	set of natural numbers	
$\mathbb{Z}$	set of integers	
$\mathbb{Q}$	set of rational numbers	
$\mathbb{R}$	set of real numbers	
$\overline{\mathbb{R}}$	set of extended real numbers	$\mathbb{R} \cup \{-\infty, \infty\}$
$[n]$	set of natural numbers less than n	$\{0, \dots, n-1\}$
$[\![\phi]\!]$	Iverson bracket of ϕ	
$\mathbb{E}\, X$	expected value of X	
$\operatorname{cl} A$	topological closure of A	
$\operatorname{int} A$	topological interior of A	
∂A	topological boundary of A	$(\operatorname{cl} A) \setminus (\operatorname{int} A)$
$\mathbb{B}_r(x)$	open ball of radius r centered at x	$\{y \mid d(x,y) < r\}$
$\mathcal{L}(V)$	space of bounded linear operators on V	

2.2 Set theory

Definition 1. The **Cartesian product** of an indexed family of sets is defined by

$$\Pi f = \{g : \operatorname{dom} f \to \cup \operatorname{im} f \mid \forall x \in \operatorname{dom} g, g(x) \in f(x)\} \tag{2.1}$$

Definition 2. Let $\{X_i\}_{i \in I}$ be an indexed family of sets. For any $i \in I$, the **projection map** onto i is the map $\pi_i : \prod_{i \in I} X_i \to X_i$ such that $\pi_i(x) = x_i$.

Definition 3. Let $\{X_i\}_{i \in I}$ be an indexed family of sets. For any $J \subseteq I$, the **canonical projection map** onto J is the map $\pi_J : \prod_{i \in I} X_i \to \prod_{j \in J} X_j$ such that $\pi_J(\{x_i\}_{i \in I}) = \{x_j\}_{j \in J}$.

Definition 4. Let $\{x_i\}_{i \in I}$ be an indexed family. Then x_{-i} denotes $x|_{I \setminus \{i\}}$.

Definition 5. A **set family** is a tuple (X, F) in which X is a set and $F \subseteq \mathcal{P}(X)$. That is, $\forall A \in F, A \subseteq X$.

Definition 6. A **ring of sets** is a set family closed under binary union and relative complement. That is, R is a ring of sets iff $\forall A, B \in R, A \cup B \in R \wedge A \setminus B \in R$.

Definition 7. If A is a set, the **indicator** function 1_A is the function defined by

$$1_A(x) = [\![x \in A]\!] \tag{2.2}$$

Definition 8. A **preorder** is a binary relation that is reflexive and transitive.

Definition 9. Let $\preceq$ be a preorder on a set P. Let $S \subseteq P$ and $s \in P$. s is a **supremum** or **least upper bound** of S iff $\forall u \in P(s \preceq u \leftrightarrow \forall x \in S, x \preceq u)$. s is an **infimum** or **greatest lower bound** of S iff it is a supremum under the reverse preorder $\succeq$.

2.3 Inner product, norm, and metric

Let K be a subring of $\mathbb{C}$ that is closed under complex conjugation. Let V be a module over K.

Definition 10. An **inner product** is a function $\langle \cdot, \cdot \rangle : V \times V \to K$ that satisfies the following properties for all $x, y, z \in V$ and $\alpha \in K$:

1. Conjugate symmetry: $\langle x, y \rangle = \langle y, x \rangle^*$.
2. Linearity: $\langle \alpha x + y, z \rangle = \alpha \langle x, z \rangle + \langle y, z \rangle$.
3. Positive-definiteness: $\langle x, x \rangle \geq 0$, with equality iff $x = 0$.

Proposition 1. The **Cauchy–Schwarz inequality** (Cauchy, 1821b; Bunyakovsky, 1859; Schwarz, 1890) states that for all $x, y \in V$:

$$|\langle x, y \rangle|^2 \leq \langle x, x \rangle \langle y, y \rangle \tag{2.3}$$

Here, $|\cdot|$ denotes the complex modulus on $\mathbb{C}$.

Definition 11. A **norm** is a function $\| \cdot \| : V \to \mathbb{R}$ that satisfies the following properties for all $x, y \in V$ and $\alpha \in K$:

1. Positive-definiteness: $\|x\| \geq 0$, with equality iff $x = 0$.
2. Absolute homogeneity: $\|\alpha x\| = |\alpha| \|x\|$.
3. Triangle inequality: $\|x + y\| \leq \|x\| + \|y\|$.

Definition 12. The **induced norm** of an inner product $\langle \cdot, \cdot \rangle$ is $\|x\| = \sqrt{\langle x, x \rangle}$. The triangle inequality follows from the Cauchy–Schwarz inequality.

Definition 13. Let X be a set. A **metric** on X is a function $d : X \times X \to \mathbb{R}$ that satisfies the following properties for all $x, y, z \in X$:

1. Positive-definiteness: $d(x, y) \geq 0$, with equality iff $x = y$.
2. Symmetry: $d(x, y) = d(y, x)$.
3. Triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$.

Definition 14. The **induced metric** of a norm $\| \cdot \|$ is $d(x, y) = \|x - y\|$.

Definition 15. A **Cauchy sequence** (Cauchy, 1821a) is a sequence $\{x_n\}_{n \in \mathbb{N}}$ such that

$$\forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall i, j > N, d(x_i, x_j) < \varepsilon \tag{2.4}$$

Definition 16. A **complete space** is a space in which every Cauchy sequence has a limit.

Definition 17. A **Banach space** is a complete normed vector space.

2.4 Topology

Definition 18. A **topological space** is a set family (X, τ) where τ is closed under finite intersections and arbitrary unions. Elements of τ are called **open sets**.

Definition 19. Let (X, τ_X) and (Y, τ_Y) be topological spaces. A function $f : X \to Y$ is **continuous** iff the preimage of every open set is an open set. That is,

$$\forall U \in \tau_Y, f^{-1}[U] \in \tau_X \tag{2.5}$$

Definition 20. Let $\{(X_i, \tau_i)\}_{i \in I}$ be a collection of topological spaces. The **product topology** is the smallest topology on ΠX such that every projection map is continuous.

Proposition 2. Under the product topology, for any topological space Y, a function $f : Y \to \prod_{i \in I} X_i$ is continuous iff, for each $i \in I$, the component function $\pi_i \circ f$ is continuous.

Definition 21. Let (X, τ) be a topological space. A **base** (or basis) for τ is a family $\mathcal{B} \subseteq \tau$ of open sets such that every open set in the topology can be represented as the union of some subfamily of $\mathcal{B}$:

$$(\forall U \in \tau)(\exists \mathcal{F} \subseteq \mathcal{B})(\cup \mathcal{F} = U) \tag{2.6}$$

Definition 22. Let X be a set. A family $\mathcal{B}$ of subsets of X satisfies the **base axioms** (and thus forms a valid base for a topology on X) iff it satisfies the following two conditions:

1. **Covering:** The union of all sets in $\mathcal{B}$ is the entire set X.

$$\bigcup \mathcal{B} = X \tag{2.7}$$

   Equivalently, for every $x \in X$, there exists at least one $B \in \mathcal{B}$ such that $x \in B$.

2. **Intersection:** If $B_1, B_2 \in \mathcal{B}$ and x is a point in their intersection ($x \in B_1 \cap B_2$), then there exists a third basis set $B_3 \in \mathcal{B}$ that contains x and fits entirely inside the intersection:

$$\exists B_3 \in \mathcal{B} : x \in B_3 \subseteq B_1 \cap B_2 \tag{2.8}$$

Definition 23. Let X be a set and let $\mathcal{B}$ be a family of subsets of X that satisfies the base axioms. The **topology generated** by $\mathcal{B}$ is the collection of all unions of subfamilies of $\mathcal{B}$:

$$\tau = \left\{ \bigcup \mathcal{F} \;\middle|\; \mathcal{F} \subseteq \mathcal{B} \right\} \tag{2.9}$$

Definition 24. A **topological field** is a field equipped with a topology such that all field operations (addition, multiplication, and inversion on non-zero elements) are continuous.

Definition 25. A **topological vector space** (TVS) is a vector space over a topological field equipped with a topology such that the vector addition and scalar multiplication maps are continuous.

Definition 26. Let (X, τ) be a topological space. Let $x \in X$ be a point. An **open neighborhood** of x is an open set $U \in \tau$ such that $x \in U$.

Definition 27. A **locally convex topological vector space** (LCTVS) is a TVS in which every open neighborhood of the zero vector contains a convex open neighborhood of the zero vector. That is, it is a TVS that possesses a local base at the zero vector consisting entirely of convex sets.

Definition 28. Two points $x, y \in X$ can be **separated** by open neighborhoods iff there exist open sets $U, V \in \tau$ that are disjoint ($U \cap V = \varnothing$) and $x \in U, y \in V$.

Definition 29. A topological space **separates points** (equivalently, is a **Hausdorff space** or T_2 space) iff any two distinct points can be separated by open neighborhoods.

2.4.1 Euclidean space

Definition 30. The **Euclidean inner product** is

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i \tag{2.10}$$

Definition 31. The **Euclidean norm** is the norm induced by this inner product.

Definition 32. The **Euclidean metric** is the metric induced by this norm.

Definition 33. The **Euclidean basis** is the collection of open balls induced by this metric:

$$B = \{\mathbb{B}_r(x) \mid x \in \mathbb{R}^n \wedge r \in \mathbb{R} \wedge r > 0\} \tag{2.11}$$

Definition 34. The **Euclidean topology** is the topology generated by this basis.

2.5 Measure theory

2.5.1 Sigma algebra

Definition 35. Let X be a set. A **σ-algebra** is a collection $\Sigma \subseteq \mathcal{P}(X)$ where $X \in \Sigma$ and Σ is closed under complements and countable unions.

Definition 36. A **measurable space** is a pair (X, Σ) where Σ is a σ-algebra on X. Elements of Σ are called **measurable sets**.

Definition 37. Let (X, Σ_X) and (Y, Σ_Y) be measurable spaces. A function $f : X \to Y$ is **measurable** iff the preimage of every measurable set is a measurable set. That is,

$$\forall E \in \Sigma_Y, f^{-1}[E] \in \Sigma_X \tag{2.12}$$

Definition 38. Let $\{(X_i, \Sigma_i)\}_{i \in I}$ be a collection of measurable spaces. The **product σ-algebra** is the smallest σ-algebra on ΠX such that every projection map is measurable. It is denoted by $\bigotimes_{i \in I} \Sigma_i$.

Proposition 3. Under the product σ-algebra, for any measurable space Y, a function $f : Y \to \prod_{i \in I} X_i$ is measurable iff, for each $i \in I$, the component function $\pi_i \circ f$ is measurable.

Definition 39. Let X be a set and (Y, Σ) be a measurable space. For each $x \in X$, the evaluation functional $e_x \in Y^X \to Y$ is defined by $e_x(f) = f(x)$. The **σ-algebra generated by the evaluation functionals** is the product σ-algebra on Y^X.

Definition 40. Let (X, F) be a set family. The σ-algebra generated by F is the smallest σ-algebra that includes F:

$$\sigma(F) = \cap\{\Sigma \subseteq \mathcal{P}(X) \mid F \subseteq \Sigma \wedge \Sigma \text{ is a } \sigma\text{-algebra}\} \tag{2.13}$$

Definition 41. Let (X, τ) be a topological space. The **Borel σ-algebra** is the σ-algebra generated by τ: $\mathcal{B}(X) = \sigma(\tau)$. Elements of $\mathcal{B}(X)$ are called **Borel sets**.

To equip sets with σ-algebras, we adopt the following conventions.

- **Topological spaces:** If X is a topological space, it is equipped with the Borel σ-algebra $\mathcal{B}(X)$.
- **Banach spaces:** If V is a Banach space, it is equipped with the Borel σ-algebra generated by its norm topology.
- **Product spaces:** If $\{(X_i, \Sigma_i)\}_{i \in I}$ is a collection of measurable spaces, the Cartesian product $\prod_{i \in I} X_i$ is equipped with the product σ-algebra.
- **Sets of measures:** If (X, Σ) is a measurable space, any set of measures is equipped with the σ-algebra generated by the evaluation functionals.

2.5.2 Measure

Definition 42. Let R be a ring of sets. A **pre-measure** on R is a function $\nu : R \to [0, \infty]$ such that $\nu(\varnothing) = 0$ and ν is countably additive.

Definition 43. A **measure** is a **pre-measure** on a σ-algebra.

Definition 44. A **measure space** is a triple (X, Σ, μ) where μ is a measure on Σ.

Definition 45. A **probability measure** is a measure μ where $\mu(X) = 1$.

Definition 46. A **probability space** is a measure space with a probability measure.

Definition 47. A **Borel probability measure** on X is a probability measure on $(X, \mathcal{B}(X))$.

Definition 48. The set of Borel probability measures on X is denoted by $\triangle X$.

Definition 49. The **Dirac measure** at x is denoted by δ_x. It is defined by $\delta_x(A) = [\![x \in A]\!]$.

Definition 50. Let (X, Σ_X, μ) be a measure space, (Y, Σ_Y) be a measurable space, and $f : X \to Y$ be a measurable function. The **pushforward measure** of μ by f, denoted by $f_*\mu$, is the measure on (Y, Σ_Y) defined by

$$\forall B \in \Sigma_Y, f_*\mu(B) = \mu(f^{-1}[B]) \tag{2.14}$$

Definition 51. The **Carathéodory extension** of a pre-measure ν is the restriction of the outer measure μ^* of ν to the σ-algebra of μ^*-measurable sets.

Definition 52. Let $\{(X_i, \Sigma_i, \mu_i)\}_{i \in I}$ be a collection of measure spaces. The **product measure** $\bigotimes_{i \in I} \mu_i$ is the Carathéodory extension of the set function ν such that for all $\{E_i\}_{i \in I}$ where $E_i \in \Sigma_i$ for all i and $E_i = X_i$ for all but finitely many i,

$$\nu\left(\prod_{i \in I} E_i\right) = \prod_{i \in I} \mu_i(E_i) \tag{2.15}$$

Definition 53. Let $\{(X_i, \Sigma_i, \mu_i)\}_{i \in I}$ be a countable collection of measure spaces. The **sum** measure $\sum_{i \in I} \mu_i$ is the measure ν on the disjoint union $\bigsqcup_{i \in I} X_i$ such that

$$\nu(E) = \sum_{i \in I} \mu_i(E \cap X_i) \tag{2.16}$$

Definition 54. Let (X, Σ, μ) be a measure space and $A \in \Sigma$ be a measurable set. The **restriction measure** $\mu|_A \in \Sigma \to [0, \infty]$ is a new measure defined by

$$\forall B \in \Sigma, \mu|_A(B) = \mu(A \cap B) \tag{2.17}$$

Definition 55. Let $\{(X_i, \Sigma_i, \mu_i)\}_{i \in I}$ be a collection of measure spaces. A **coupling** is a measure γ on the product measurable space $(\prod_{i \in I} X_i, \bigotimes_{i \in I} \Sigma_i)$ such that the **marginal** measure along every i-th coordinate space is μ_i:

$$\forall i \in I, (\pi_i)_* \gamma = \mu_i \tag{2.18}$$

where $\pi_i \in \prod_{j \in I} X_j \to X_i$ is the i-th projection map.

We also have the following definitions.

A null	$A \in \Sigma \wedge \mu(A) = 0$
B negligible	$\exists A, B \subseteq A \wedge A$ null
A co-null	$X \setminus A$ null
B co-negligible	$X \setminus B$ negligible
$\operatorname{ess\,sup} f$	$\inf\{a \in \overline{\mathbb{R}} \mid \{x \in X \mid f(x) > a\}$ negligible$\}$
$\operatorname{ess\,inf} f$	$\sup\{a \in \overline{\mathbb{R}} \mid \{x \in X \mid f(x) < a\}$ negligible$\}$
ϕ almost nowhere	$\{x \in X \mid \phi(x)\}$ negligible
ϕ almost everywhere	$\{x \in X \mid \neg\phi(x)\}$ negligible

Definition 56. An **atom** is a measurable set $A \in \Sigma$ such that $\mu(A) > 0$ and, for every measurable subset $B \subseteq A$, $\mu(B) = 0$ or $\mu(B) = \mu(A)$.

Definition 57. A measure is **atomic** iff every measurable set of positive measure contains an atom.

Definition 58. A measure is **non-atomic**, **atomless**, or **diffuse** iff it has no atoms.

Definition 59. Let (M, d) be a metric space. Let μ and ν be probability measures on M. The **Wasserstein distance** of order p between μ and ν is

$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu, \nu)} \mathop{\mathbb{E}}_{(x,y) \sim \gamma} d(x, y)^p\right)^{\frac{1}{p}} \tag{2.19}$$

where $\Gamma(\mu, \nu)$ is the set of all couplings of μ and ν.

Proposition 4. The case $p \to \infty$ is as follows.

$$W_\infty(\mu, \nu) = \lim_{p \to \infty} W_p(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \operatorname*{ess\,sup}_{(x,y)\sim\gamma} d(x, y) \tag{2.20}$$

We adopt the following canonical conventions for equipping new spaces with measures:

- **Products:** If each X_i has a measure μ_i, we equip $\prod_{i \in I} X_i$ with the product measure $\bigotimes_{i \in I} \mu_i$.
- **Mappings:** If f is a measurable function and $\operatorname{dom} f$ has a measure μ, we equip $\operatorname{cod} f$ with the pushforward measure $f_*\mu$.
- **Subsets:** If X has a measure μ and $A \subseteq X$ is a measurable subset, we equip A with the restriction measure $\mu|_A$.
- **Disjoint unions:** If I is countable and each X_i has a measure μ_i, we equip $\bigsqcup_{i \in I} X_i$ with the sum measure $\sum_{i \in I} \mu_i$.
- **Quotients:** If $\sim$ is an equivalence relation on X and X has a measure μ, we equip the quotient space $X/\sim$ with the pushforward measure $\pi_*\mu$, where $\pi : X \to X/\sim$ is the quotient map.

2.6 Integration

Definition 60. The **integral** of a function f with respect to a measure μ is denoted by $\int_{x\sim\mu} f(x)$ or $\int_\mu f$. If μ is a probability measure, we also use the notation for expectations, $\mathbb{E}_{x\sim\mu} f(x)$ or $\mathbb{E}_\mu f$.

An integral satisfies the following fundamental properties:

Proposition 5 (Indicator functions)**.** Let A be a measurable set. Then

$$\int_\mu 1_A = \mu(A) \tag{2.21}$$

Proposition 6 (Addition)**.** Let f and g be integrable functions. Then

$$\int_\mu (f + g) = \int_\mu f + \int_\mu g \tag{2.22}$$

Proposition 7 (Scalar multiplication)**.** Let f be an integrable function and α be a scalar. Then

$$\int_\mu \alpha f = \alpha \int_\mu f \tag{2.23}$$

Definition 61. A **simple function** is a function of the form

$$f = \sum_{i=1}^{n} \alpha_i 1_{A_i} \tag{2.24}$$

where $n \in \mathbb{N}$, each α_i is a scalar, and each A_i is a measurable set.

It follows from the preceding properties that any simple function, provided its constituent sets A_i have finite measure, is integrable.

There are many types of integrals in the literature, including (but not limited to):

- The Riemann integral (Riemann, 1868)
- The Lebesgue integral (Lebesgue, 1902)
- The Riemann–Stieltjes integral (Stieltjes, 1894)
- The Lebesgue–Stieltjes integral (Saks, 1937)
- The Henstock–Kurzweil integral (Kurzweil, 1957)
- The Bochner integral (Bochner, 1933)
- The Pettis integral (Pettis, 1938)
- The Dunford integral (Dunford, 1935)

Some of these integrals are **stronger** than others; that is, they can integrate a **strictly larger** class of functions. For example, in addition to linearity and the indicator property, the **Bochner integral** satisfies the following strong continuity guarantee:

Proposition 8 (Continuity)**.** Let (Ω, Σ, μ) be a measure space. Let B be a Banach space with norm $\|\cdot\|$. Let $f : \Omega \to B$. For each $n \in \mathbb{N}$, let $g_n : \Omega \to B$ be Bochner-integrable functions. Then

$$\lim_{n\to\infty} \int_{x\sim\mu} \|g_n(x) - f(x)\| = 0 \quad \Longrightarrow \quad \lim_{n\to\infty} \int_{\mu} g_n = \int_{\mu} f \tag{2.25}$$

This means that if a sequence of functions converges to f in the mean (i.e., the integral of their normed difference vanishes), then their respective integrals converge to the integral of f.

An even more generalized formulation is the **Pettis integral**, which is entirely characterized by the following property:

Proposition 9 (Duality)**.** Let (Ω, Σ, μ) be a measure space. Let V be a LCTVS with continuous dual space V^*. Let $f : \Omega \to V$ be a function. Let $v \in V$. Then

$$\left(\forall \phi \in V^* : \phi(v) = \int_{\mu} (\phi \circ f)\right) \quad \Longrightarrow \quad \int_{\mu} f = v \tag{2.26}$$

In functional analysis, this leads directly to the core property that characterizes all such vector-valued integrals.

Definition 62. The **universal property of integration** is defined as follows. Let (Ω, Σ, μ) be a measure space. Let X and Y be LCTVSs over the same scalar field. If $f : \Omega \to X$ is an integrable function, its integral is the unique vector $\int_\mu f \in X$ such that, for any continuous linear operator $\phi : X \to Y$, the composition $\phi \circ f : \Omega \to Y$ is also integrable, and

$$\phi\left(\int_{\mu} f\right) = \int_{\mu} (\phi \circ f) \tag{2.27}$$

Proposition 10. The universal property completely determines the integral. Evaluating it over the continuous dual space (by letting Y be the underlying scalar field, so that $\phi \in X^*$) uniquely isolates the vector in X, provided the topology of X separates points.

2.7 Functions

Let f be a function.

Definition 63. Let S be a set. The **restriction** of f to S is denoted by $f|_S$. It is equal to $f \cap (S \times \operatorname{im} f)$, by the standard definition of a function as a set of ordered input-output pairs.

Definition 64. The **substitution** $f[a \mapsto b]$ is the function defined by

$$f[a \mapsto b](x) = \begin{cases} b & x = a \\ f(x) & x \neq a \end{cases} \tag{2.28}$$

Definition 65. The upper-right **Dini derivative** of f at x in direction d is

$$(D^+ f)(x)(d) = \limsup_{t \downarrow 0} \frac{f(x + td) - f(x)}{t} \tag{2.29}$$

If f is differentiable, this is just $\langle \nabla f(x), d \rangle$.

Definition 66. f is **convex** iff for all $x, y \in \operatorname{dom} f$ and all $t \in [0, 1]$:

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) \tag{2.30}$$

Definition 67. f is m**-convex** iff $x \mapsto f(x) - \frac{m}{2}\|x\|^2$ is convex.

Proposition 11. If $f_t(x)$ is m_t-convex, $\int_{t \sim \mu} f_t(x)$ is $\int_{t \sim \mu} m_t$-convex.

Proposition 12. If $f_t(x)$ is m_t-convex, $\sup_{t \in T} f_t(x)$ is $\inf_{t \in T} m_t$-convex.

Definition 68. f is **strictly-convex** iff for all $x, y \in \operatorname{dom} f$ such that $x \neq y$ and all $t \in (0, 1)$:

$$f(tx + (1-t)y) < tf(x) + (1-t)f(y) \tag{2.31}$$

Definition 69. f is **quasi-convex** iff for all $x, y \in \operatorname{dom} f$ and all $t \in [0, 1]$:

$$f(tx + (1-t)y) \leq \max\{f(x), f(y)\} \tag{2.32}$$

Definition 70. f is **pseudo-convex** iff for all $x, y \in \operatorname{dom} f$:

$$(D^+ f)(x)(y - x) \geq 0 \implies f(y) \geq f(x) \tag{2.33}$$

Definition 71. f is **monotone** iff for all $x, y \in \operatorname{dom} f$:

$$\langle f(x) - f(y), x - y \rangle \geq 0 \tag{2.34}$$

Definition 72. f is m**-monotone** iff $x \mapsto f(x) - mx$ is monotone. Equivalently, for all $x, y \in \operatorname{dom} f$:

$$\langle f(x) - f(y), x - y \rangle \geq m\|x - y\|^2 \tag{2.35}$$

Definition 73. f is β**-cocoercive** iff for all $x, y \in \operatorname{dom} f$:

$$\langle f(x) - f(y), x - y \rangle \geq \beta\|f(x) - f(y)\|^2 \tag{2.36}$$

Definition 74. f is (α, C)**-Hölder continuous** iff for all $x, y \in \operatorname{dom} f$:

$$d(f(x), f(y)) \leq C d(x, y)^{\alpha} \tag{2.37}$$

Definition 75. f is C**-Lipschitz continuous** iff it is $(1, C)$-Hölder continuous.

Definition 76. Let (X, Σ, μ) be a measure space. Let $0 < p < \infty$. The L^p **quasi-norm** of a function f is

$$\|f\|_p = \left(\int_{\mu} |f|^p \right)^{\frac{1}{p}} \tag{2.38}$$

It is a norm for $p \geq 1$.

Definition 77. Likewise, the L^{∞} **norm** is the essential supremum of $|f|$:

$$\|f\|_{\infty} = \operatorname{ess\,sup} |f| \tag{2.39}$$

Functions with finite L^{∞} are also called **essentially bounded**.

Definition 78. $L^p(A; B)$ is the set of (equivalence classes of) functions from A to B with finite L^p norm.

2.8 Pseudo-gradients

Definition 79. A **pseudo-gradient** is an estimator for the gradient of some smoothed approximation of a function. It uses only function evaluations, as opposed to exact analytical methods like forward or reverse automatic differentiation.

A pseudo-gradient is typically used when the true gradient is unavailable, difficult to calculate, or when the function is non-differentiable. Consider the problem of maximizing $f : \mathbb{R}^d \to \mathbb{R}$ with access to its values but not its derivatives. This setting is called **zeroth-order optimization**. One approach to this problem computes estimates of the gradient $g(x) \approx \nabla f(x)$ and applies gradient-based optimization. The gradient can be estimated via **finite differences** as

$$g(x)_i = \tfrac{1}{\sigma}(f(x + \sigma e_i) - f(x)) \tag{2.40}$$

for all $i \in [d]$, where e_i is the i-th standard basis vector and σ is a small scalar. However, the number of queries needed scales linearly with the number of dimensions d.

To circumvent this scaling issue, an alternative approach evaluates the function at **randomly-sampled points** and estimates the gradient as a sum of directional derivatives along **random directions** (Duchi, Jordan, et al., 2015; Nesterov and Spokoiny, 2017; Shamir, 2017; Berahas et al., 2022). These methods compute an unbiased estimator of the gradient of a *smoothed* version of f, which is induced by stochastically perturbing the input under a distribution μ_1 and taking the expectation (Duchi, Bartlett, and Wainwright, 2012). Specifically, these take the form of

$$\nabla_x \mathop{\mathbb{E}}_{z \sim \mu_1} f(x + \sigma z) = \tfrac{1}{\sigma} \mathop{\mathbb{E}}_{z \sim \mu_2} f(x + \sigma z) z \tag{2.41}$$

where μ_1 and μ_2 are two (possibly different) distributions. For example, one can use the following **distributions**:

- **Gaussian smoothing**: $\mu_1 = \mu_2 = \text{Normal}(0, I_d)$.
- **Ball smoothing**: $\mu_1 = \text{Uniform}(X)$ and $\mu_2 = \text{Uniform}(\partial X)$ where X is the d-dimensional unit ball of radius $\sqrt{d}$.

In practice, we often estimate $\nabla_x \mathbb{E}_{z\sim\mu_1} f(x + \sigma z)$ by averaging *multiple* samples together:

$$\frac{1}{\sigma N} \sum_{i=1}^{N} a_i z_i \tag{2.42}$$

where $z_i \sim \mu_2$ are independent samples and a_i is evaluated using one of the following **stencils**:

- **Single-point**: $f(x + \sigma z_i)$
- **Forward-difference**: $f(x + \sigma z_i) - f(x)$
- **Central-difference**: $\frac{1}{2}(f(x + \sigma z_i) - f(x - \sigma z_i))$

Because the single-point stencil suffers from a large variance that diverges to infinity as σ approaches 0, the latter two are typically preferred in practice (Berahas et al., 2022).

These mathematical foundations underpin several classes of algorithms. Black-box zeroth-order optimization uses only function evaluations to optimize a black-box function with respect to a set of inputs. In particular, it does not require gradients. A prominent class of these algorithms is **evolution strategies** (ES) (Rechenberg, 1973; Schwefel, 1977; Rechenberg, 1978; Bäck, 1996; Bäck, Fogel, and Michalewicz, 1997; Eiben and Smith, 2003). Historically, these methods maintained and evolved a *population* of parameter vectors.

Natural evolution strategies (NES) (Wierstra et al., 2014; Yi et al., 2009) represented the population as a **distribution over parameters** and maximized its average objective value using the score function estimator. For many parameter distributions, such as Gaussian smoothing, this is equivalent to evaluating the function at randomly-sampled points and estimating the gradient as a sum of estimates of directional derivatives along random directions (Fu, 2015; Duchi, Jordan, et al., 2015; Nesterov and Spokoiny, 2017; Shamir, 2017; Berahas et al., 2022).

Salimans et al. (2017) applied Gaussian smoothing to single-agent reinforcement learning and obtained competitive results on standard benchmarks. Lenc et al. (2019) showed that ES is a viable method for learning non-differentiable parameters of large supervised models.

A closely related technique for smoothed gradient estimation is **simultaneous-perturbation stochastic approximation** (SPSA). Spall (1992) introduced this method, which perturbs each coordinate with Rademacher variates $\mu_1 = \mu_2 = \text{Uniform}(\{-1, 1\}^d)$ and uses the central-difference stencil. A Taylor expansion of f shows that this is a good estimate of the true gradient when σ is small. Later, Spall (1997) introduced a one-measurement variant of SPSA that utilized the single-point stencil.

Chapter 3

Game theory definitions

In this section, we define some fundamental concepts in game theory. These include various types of games, as well as solution concepts and performance metrics. We also give a mathematically rigorous formulation of the setting we are tackling.

3.1 Normal-form game

von Neumann (1928) introduced the concept of a normal-form game. The motivation was to eliminate the chaotic complexities of time, sequence, and hidden information in strategic interactions. Von Neumann later expanded on this concept, and game theory in general, alongside economist Oskar Morgenstern in their seminal book, *Theory of Games and Economic Behavior* (von Neumann and Morgenstern, 1947).

Definition 80. A **normal-form game** (NFG) is a tuple (I, S, u) whose elements are as follows.

- I is a set of **players**.
- S_i is a set of **strategies** for each player i. A **strategy profile** is a map from each player to an element of its strategy set. That is, it is an element of ΠS.
- $u \in \Pi S \times I \to \mathbb{R}$ is a **utility function**. It maps a strategy profile and player to a utility.

Definition 81. A **best response** (BR) is a strategy that maximizes a player's utility given the other players' strategies. The set of best responses to strategy profile s by player i is denoted by

$$\mathrm{BR}(s)_i = \operatorname*{argmax}_{x \in S_i} u(s[i \mapsto x])_i \tag{3.1}$$

Definition 82. A **Nash equilibrium** (NE) is a strategy profile for which each player's strategy is a best response to the other players' strategies. Thus

$$s \in \mathrm{NE} \quad \iff \quad \forall i \in I, s_i \in \mathrm{BR}(s)_i \tag{3.2}$$

Definition 83. A player's **BR value** is the supremum utility it could attain by unilaterally changing its strategy. It is denoted by

$$\mathrm{BRV}(s)_i = \sup_{x \in S_i} u(s[i \mapsto x])_i \tag{3.3}$$

Definition 84. A player's **regret** is the supremum utility it could gain by unilaterally changing its strategy. It is denoted by

$$\operatorname{Reg}(s)_i = \operatorname{BRV}(s)_i - u(s)_i \tag{3.4}$$

Definition 85. An **ε-Nash equilibrium** (ε-NE) is a strategy profile for which each player's regret is at most ε. Thus

$$s \in \varepsilon\text{-NE} \iff \sup_{i \in I} \operatorname{Reg}(s)_i \leq \varepsilon \tag{3.5}$$

The special case $\varepsilon = 0$ coincides with NE.

In the following, let μ be a measure on I.

Definition 86. The **social welfare** of a strategy profile is the aggregate utility across players. It is denoted by

$$\operatorname{Wel}(s) = \int_{i \sim \mu} u(s)_i \tag{3.6}$$

Definition 87. The **exploitability** of a strategy profile is the aggregate regret across players. It is denoted by

$$\operatorname{Expl}(s) = \int_{i \sim \mu} \operatorname{Reg}(s)_i \tag{3.7}$$

Exploitability is a standard metric of "closeness" to NE in the literature (Lanctot, Zambaldi, et al., 2017; Lockhart et al., 2019; Walton and Lisy, 2021; Timbers et al., 2022). It is sometimes called "NashConv".

Definition 88. The **Nikaido–Isoda function** (NI) is

$$\operatorname{NI}(x, y) = \int_{i \sim \mu} u(x[i \mapsto y_i])_i - u(x)_i \tag{3.8}$$

The Nikaido–Isoda function was introduced by Nikaido and Isoda (1955), and studied by Flåm and Antipin (1996), Flåm and Ruszczyński (2008), and Hou, Wen, and Chang (2018).

Note that

$$\operatorname{Expl}(x) = \sup_{y \in \Pi S} \operatorname{NI}(x, y) \tag{3.9}$$

Therefore, finding an NE (if one exists) is equivalent to solving the min-max problem $\inf_x \sup_y \operatorname{NI}(x, y)$. Some prior work has used this function to search for NE (Uryasev and Rubinstein, 1994; Berridge and Krawczyk, 1997; Krawczyk and Uryasev, 2000; Krawczyk, 2005; Flåm and Ruszczyński, 2008; Gürkan and Pang, 2009; Heusinger and Kanzow, 2009a; Heusinger and Kanzow, 2009b; Qu and Zhao, 2013; Hou, Wen, and Chang, 2018; Raghunathan, Cherian, and Jha, 2019; Tsaknakis and Hong, 2021).

Definition 89. An **exact potential** is a function $\phi : \Pi S \to \mathbb{R}$ such that

$$u(s[i \mapsto x])_i - u(s)_i = \phi(s[i \mapsto x]) - \phi(s) \tag{3.10}$$

for all $s \in \Pi S, i \in I, x \in S_i$. That is, a deviation increases utility by the amount that it increases the potential.

Definition 90. An **ordinal potential** is a function $\phi : \Pi S \to \mathbb{R}$ such that

$$u(s[i \mapsto x])_i > u(s)_i \quad \leftrightarrow \quad \phi(s[i \mapsto x]) > \phi(s) \tag{3.11}$$

for all $s \in \Pi S, i \in I, x \in S_i$. That is, a deviation increases utility iff it increases the potential.

Proposition 13. A **global maximum of a potential** is an NE.

Proof. Suppose $s \in \operatorname{argmax} \phi$. For every $i \in I$ and $x \in S_i$, $\phi(s[i \mapsto x]) \leq \phi(s)$, and thus $u(s[i \mapsto x])_i \leq u(s)_i$. Thus, for every $i \in I$, $s_i \in \mathrm{BR}(s)_i$. Thus $s \in \mathrm{NE}$. □

Proposition 14. If the utility function is **twice-differentiable**, an exact potential function exists iff

$$\frac{\partial^2 u(s)_i}{\partial s_i \partial s_j} = \frac{\partial^2 u(s)_j}{\partial s_i \partial s_j} \tag{3.12}$$

for all $i, j \in I$. That is, the marginal effect of a player's strategy on another player's marginal utility equals the marginal effect of the latter's strategy on the former. This theorem is due to Monderer and Shapley (1996).

Definition 91. Let V be a topological vector space. Suppose $\forall i \in I, S_i \subseteq V$. An **aggregative game** (Selten, 1970; Cornes and Hartley, 2007) is a game such that there exists a measure μ on I and functions $f_i : S_i \times V \to \mathbb{R}$ for every $i \in I$ such that

$$\bar{s} = \int_{j \sim \mu} s_j \tag{3.13}$$

$$u(s)_i = f_i(s_i, \bar{s}) \tag{3.14}$$

for every $i \in I$. Here, $\bar{s}$ is the **aggregate strategy** across players. In other words, each player's utility depends on its own strategy and on an aggregate of all players' strategies.

3.2 Mixed-strategy game

Definition 92. A **mixed-strategy game** is a tuple (I, A, R) whose elements are as follows.

- I is a set of **players**.
- A_i is a set of **actions** for each player i.
- $R : \Pi A \to \mathbb{R}$ is a **reward function**.

Definition 93. A **mixed strategy** for player i is a probability measure $\sigma_i \in \triangle A_i$.

Definition 94. The **joint mixed strategy** is the product measure $\sigma \in \triangle \Pi A$ defined by

$$\sigma(a) = \bigotimes_{i \in I} \sigma_i(a_i) \tag{3.15}$$

Definition 95. The **utility** is the expected reward under sampling of an action profile:

$$u(\sigma) = \mathop{\mathbb{E}}_{a \sim \sigma} R(a) \tag{3.16}$$

Definition 96. The **induced NFG** is $\left(I, \prod_{i \in I} \triangle A_i, u\right)$. That is, each player chooses a mixed strategy, and the utility is the expected utility under the resulting joint mixed strategy.

Definition 97. A **mixed strategy Nash equilibrium** (MSNE) of the NFG (I, A, R) is an NE of the aforementioned induced NFG.

Definition 98. A **pure strategy Nash equilibrium** (PSNE) is an MSNE where each factor distribution is a Dirac delta function.

Proposition 15. A player's BR value can be computed by maximizing over actions:

$$\mathrm{BRV}(\sigma)_i = \sup_{x \in \triangle A_i} u(\sigma[i \mapsto x])_i = \sup_{x \in A_i} u(\sigma[i \mapsto \delta_x])_i \tag{3.17}$$

Definition 99. A **correlated equilibrium** (CE) (Aumann, 1974) is a distribution $\sigma \in \triangle \Pi A$ over action profiles such that

$$\mathop{\mathbb{E}}_{a \sim \sigma} u(a)_i \geq \mathop{\mathbb{E}}_{a \sim \sigma} u(a[i \mapsto f_i(a_i)])_i \tag{3.18}$$

for every player $i \in I$ and deviation function $f_i : A_i \to A_i$. Unlike NE, which has to be a product distribution, this distribution can be correlated.

Definition 100. A **coarse correlated equilibrium** (CCE) (Moulin and Vial, 1978) is a distribution $\sigma \in \triangle \Pi A$ over action profiles such that

$$\mathop{\mathbb{E}}_{a \sim \sigma} u(a)_i \geq \mathop{\mathbb{E}}_{a \sim \sigma} u(a[i \mapsto a_i'])_i \tag{3.19}$$

for every player $i \in I$ and deviation action $a_i' \in A_i$.

Proposition 16. The previous solution concepts are related as follows.

$$\text{PSNE} \subseteq \text{MSNE} \subseteq \text{CE} \subseteq \text{CCE} \tag{3.20}$$

3.3 Bayesian game

Harsanyi (1967) introduced the concept of a Bayesian game. The motivation was to formalize the concept of **incomplete information**, where players are uncertain about the rules, payoffs, or others' available moves. **Harsanyi's transformation** converts an incomplete information game into a complete but imperfect information game. It does this as follows. There is a move by Nature at the start of the game. Nature randomly assigns each player a "type" or observation, representing their private information (e.g., cost or preference). Players know their own type but only have a probability distribution (beliefs) over others' types. By framing the uncertainty as a move by Nature, this allowed existing techniques and tools for NE to be used.

Definition 101. A **Bayesian game** is a tuple (I, S, O, A, Z, R, ρ) whose elements are as follows. We assume all sets are equipped with σ-algebras (as described in Section 2.5.1) and all functions are measurable.

- I is a set of **players**.

- S is a set of **states**.
- O_i is a set of **observations** for each player i.
- A_i is a set of **actions** for each player i.
- $Z : S \to \triangle\Pi O$ is an **observation function**.
- $R : S \times \Pi A \to \mathbb{R}^I$ is a **reward function**.
- $\rho \in \triangle S$ is a **state distribution**.

Definition 102. A **policy** for player i is a function $\pi_i : O_i \to \triangle A_i$. It maps each observation to a distribution of actions.

Definition 103. The **joint policy** is the function $\pi : \Pi O \to \triangle\Pi A$ defined by

$$\pi(a \mid o) = \bigotimes_{i \in I} \pi_i(a_i \mid o_i) \tag{3.21}$$

Definition 104. The **utility** is the expected reward under the following process:

$$u(\pi) = \mathop{\mathbb{E}}_{s\sim\rho} \mathop{\mathbb{E}}_{o\sim Z(s)} \mathop{\mathbb{E}}_{a\sim\pi(o)} \mathop{\mathbb{E}}_{r\sim R(s,a)} r \tag{3.22}$$

That is, a state is sampled, a joint observation is sampled given the state, a joint action is sampled given the observation, and a joint reward is sampled given the state and joint action.

Proposition 17. A player's BR value can be computed as follows:

$$\begin{aligned}
\mathrm{BRV}(\pi)_i &= \sup_{x : O_i \to \triangle A_i} u(\pi[i \mapsto x])_i && (3.23)\\
&= \sup_{x : O_i \to \triangle A_i} \mathop{\mathbb{E}}_{s\sim\rho} \mathop{\mathbb{E}}_{o\sim Z(s)} \mathop{\mathbb{E}}_{a\sim\pi[i\mapsto x](o)} \mathop{\mathbb{E}}_{r\sim R(s,a)} r_i && (3.24)\\
&= \mathop{\mathbb{E}}_{o_i} \sup_{x \in \triangle A_i} \mathop{\mathbb{E}}_{a_i\sim x} \mathop{\mathbb{E}}_{s|o_i} \mathop{\mathbb{E}}_{a_{-i}|s} \mathop{\mathbb{E}}_{r\sim R(s,a)} r_i && (3.25)\\
&= \mathop{\mathbb{E}}_{o_i} \sup_{a_i \in A_i} \mathop{\mathbb{E}}_{s|o_i} \mathop{\mathbb{E}}_{a_{-i}|s} \mathop{\mathbb{E}}_{r\sim R(s,a)} r_i && (3.26)
\end{aligned}$$

This means that, instead of maximizing over the policy space $O_i \to \triangle A_i$, we can sample observations, maximize over the action space A_i, and sample states conditioned on the observation. This technique was used by Bichler, Fichtl, Heidekrüger, et al. (2021, Supplementary Section 3) to estimate exploitabilities.

3.4 Mean-field game

Definition 105. A **mean-field game** (MFG) is a tuple $(X, A, T, R, \rho, \gamma)$ whose elements are as follows. We assume that all sets are equipped with σ-algebras (as described in Section 2.5.1) and all functions are measurable.

- X is a set of **states** that an individual player can occupy.
- A is a set of **actions** that an individual player can perform.

- $T : X \times M(X) \times A \to \triangle X$ is a **transition function**.
- $R : X \times M(X) \times A \to \mathbb{R}$ is a **reward function**.
- $\rho \in M(X)$ is an **initial state distribution**.
- $\gamma \in \mathbb{R}$ is a **discount factor**.

Here, $M(X)$ is the set of measures on X. A **population distribution** is an element of $M(X)$, representing a distribution of players over the set of states X, i.e., an occupancy measure for the states.

A **Markov policy** is a sequence $\{\pi_t\}_{t\in\mathbb{N}}$ of stochastic kernels on A given X.

A **population flow** is a sequence $\{m_t\}_{t\in\mathbb{N}}$ of measures on X.

If a player executes a Markov policy π under an exogenous population flow m, its individual **trajectory** is generated as follows:

$$x_0 \sim \rho \tag{3.27}$$
$$a_t \sim \pi_t(x_t) \tag{3.28}$$
$$r_t = R(x_t, m_t, a_t) \tag{3.29}$$
$$x_{t+1} \sim T(x_t, m_t, a_t) \tag{3.30}$$

The player's **expected return** (cumulative reward) is

$$J(\pi, m) = \mathbb{E} \sum_{t=0}^{\infty} \gamma^t r_t \tag{3.31}$$

Equivalently,

$$q_t(x, a) = R(x, m_t, a) + \gamma \mathop{\mathbb{E}}_{x'\sim T(x,m_t,a)} v_{t+1}(x') \tag{3.32}$$
$$v_t(x) = \mathop{\mathbb{E}}_{a\sim\pi_t(x)} q_t(x, a) \tag{3.33}$$
$$J(\pi, m) = \mathop{\mathbb{E}}_{x\sim\rho} v_0(x) \tag{3.34}$$

If a **representative player** follows a Markov policy π, the **induced population flow** $m = \Lambda(\pi)$ is

$$m_0 = \rho \tag{3.35}$$
$$m_{t+1}(B) = \int_{x_t\sim m_t} \mathop{\mathbb{E}}_{a_t\sim\pi_t(x_t)} T(x_t, m_t, a_t)(B) \tag{3.36}$$

A pair (π, m) consisting of a Markov policy π and population flow m is a **mean-field equilibrium** iff the following two conditions hold:

$$\pi \in \operatorname*{argmax}_{\pi} J(\pi, m) \qquad \text{rationality} \tag{3.37}$$
$$m = \Lambda(\pi) \qquad \text{self-consistency} \tag{3.38}$$

3.5 Partially observable stochastic game

Definition 106. A **partially observable stochastic game** (POSG) is a tuple $(I, S, O, A, Z, T, V, R, \Gamma, \rho)$ whose elements are as follows. We assume that all sets are equipped with σ-algebras (as described in Section 2.5.1) and all functions are measurable.

- I is a set of **players**.
- S is a set of **states**.
- O_i is a set of **observations** for each player i.
- A_i is a set of **actions** for each player i.
- $Z : S \to \triangle \Pi O$ is an **observation function**.
- $T : S \times \Pi A \to \triangle S$ is a **transition function**.
- V is a Banach space of **value vectors**, which are functions in $I \to \mathbb{R}$ that are essentially bounded (c.f. Section 2.7) and measurable (c.f. Section 2.5.1).
- $R : S \times \Pi A \times S \to \triangle V$ is a **reward function**.
- $\Gamma : S \times \Pi A \times S \to \triangle \mathcal{L}(V)$ is a **discount function**.
- $\rho \in \triangle S$ is an **initial state distribution**.

Notice that a discount factor is a linear operator on the space of value vectors.[1]

Definition 107. A **policy** for player i is a function $\pi_i : O_i \to \triangle A_i$. It maps each observation to a distribution of actions. For simplicity of exposition, the policies are assumed to be *reactive*, that is, depend only on the current observation. This models imperfect recall games with no memory. However, this can be extended to policies *with* memory.

[1] Many linear operators can be expressed as **kernels**, that is, as integrals. However, in infinite spaces, there are more exotic operators that cannot be expressed in this way. An example is the limit operator. Suppose the set of players is $\mathbb{N}$. Consider the linear operator γ such that Player 0's value depends on the long-term trend of the entire population, regardless of any finite group of individuals:

$$(\gamma v)(0) = \lim_{n \to \infty} v(n) \tag{3.39}$$

(Technically, since the limit does not always exist, we extend this to a **Banach limit**, which is a non-negative, shift-invariant, linear functional that extends the usual limit and assigns a limit-like value to all bounded sequences.) This is a linear operator, but it cannot be represented as a kernel. Conceptually, the "mass" of this "measure" is not located on any specific integer, nor spread out as a density. Instead, its mass is concentrated "at infinity"—more precisely, in the so-called **Stone-Čech compactification** of $\mathbb{N}$. Informally, the Stone-Čech compactification of a topological space captures the "ways of going to infinity" in that space and adjoins them to it.

Kernel operators cover most cases in physics and economics (local interactions, weighted averages, and mean fields). Exotic operators (like the Banach limit) represent "ghost" interactions with the infinite tail of the population that cannot be localized. More precisely, the Yosida-Hewitt decomposition (Yosida and Hewitt, 1952) shows that a finitely-additive measure can be uniquely decomposed into the sum of a countably additive measure (the kernel component, in which the mass is distributed across points or "small" sets) and a "purely" finitely additive measure (the exotic component, in which the mass exists at the "boundary" or "infinity").

Exotic operators can model esoteric phenomena like "rational bubbles" in infinite economies where value comes from "infinity" rather than any fundamental agent (Bewley, 1972; Gilles and LeRoy, 1992). Under the **Yosida-Hewitt decomposition**, the price of an asset over an infinite horizon decomposes into the sum of the **fundamental value** and the **rational bubble**. The fundamental value is the standard discounted sum of all future dividends paid to specific, localized agents in time. The rational bubble is a component of the price that exists purely because agents believe they can sell the asset to someone else in the future, *ad infinitum*. It has been argued that **fiat money** is an example of a rational bubble (Gilles and LeRoy, 1992; Santos and Woodford, 1997; Huang and Werner, 2000).

Definition 108. The **joint policy** is the function $\pi : \Pi O \to \triangle \Pi A$ defined by

$$\pi(a \mid o) = \bigotimes_{i \in I} \pi_i(a_i \mid o_i) \tag{3.40}$$

3.5.1 Episode formulation

Definition 109. An **episode** is a tuple (s, o, a, r, γ) where $s_0 \sim \rho$ and, for each timestep $t \in \mathbb{N}$,

$$o_t \sim Z(s_t) \tag{3.41}$$
$$a_t \sim \pi(o_t) \tag{3.42}$$
$$s_{t+1} \sim T(s_t, a_t) \tag{3.43}$$
$$r_t \sim R(s_t, a_t, s_{t+1}) \tag{3.44}$$
$$\gamma_t \sim \Gamma(s_t, a_t, s_{t+1}) \tag{3.45}$$

Definition 110. The **return** for timestep t is

$$g_t = r_t + \gamma_t g_{t+1} \tag{3.46}$$

Definition 111. The **cumulative discount** from timestep i to timestep $j \geq i$ is

$$\rho_{ij} = \prod_{k=i}^{j-1} \gamma_k \tag{3.47}$$

Thus the return for timestep t is

$$g_i = \sum_{j=i}^{\infty} \rho_{ij} r_j \tag{3.48}$$

Definition 112. The **utility** is

$$u(\pi) = \mathbb{E}\, g_0 \tag{3.49}$$

Definition 113. The **induced NFG** is $\left(I, \prod_{i \in I}(O_i \to \triangle A_i), u\right)$. That is, each player chooses a policy, and the utility is the expected utility under the resulting joint policy.

3.5.2 Value formulation

An equivalent way to formulate the expected utility is as follows. Given a joint policy π, let

- $q(s, a)$ be the value of state s and action a.
- $v(s)$ be the value of state s.
- $u(\pi)$ be the utility.

Then

$$q(s, a) = \mathop{\mathbb{E}}_{s' \sim T(s,a)} \mathop{\mathbb{E}}_{r \sim R(s,a,s')} \mathop{\mathbb{E}}_{\gamma \sim \Gamma(s,a,s')} r + \gamma v(s') \tag{3.50}$$
$$v(s) = \mathop{\mathbb{E}}_{o \sim Z(s)} \mathop{\mathbb{E}}_{a \sim \pi(o)} q(s, a) \tag{3.51}$$
$$u(\pi) = \mathop{\mathbb{E}}_{s \sim \rho} v(s) \tag{3.52}$$

where q and v are defined recursively in terms of each other.

Chapter 4

Prior work

In this section, we describe prior work in each of the main areas we are tackling.

4.1 Prior work on infinite-action games

Ganzfried and Sandholm (2010) presented a procedure for solving large imperfect-information games by solving an infinite approximation of the original game and mapping the equilibrium to a strategy profile in the finite game. Perhaps counterintuitively, this infinite approximation could often be solved much more easily than the original finite game. To find an equilibrium in continuous games, the algorithm exploited a qualitative model of the equilibrium structure as an additional input.

4.1.1 Action space discretization

The typical approach to computing an equilibrium of a game with continuous action spaces involves discretizing the action space. Building on this approach, Bichler, Fichtl, and Oberlechner (2023) introduced an algorithmic framework for Bayesian games with continuous type and action spaces. Specifically, they discretized the type and action spaces and implemented gradient dynamics in the discretized version of the game without using neural networks. They computed distributional strategies (Milgrom and Weber, 1985) (a form of mixed strategies for Bayesian games) via online convex optimization, namely **simultaneous online dual averaging** (SODA). Consequently, they demonstrated that the equilibrium of their discretized game approximated an equilibrium in the continuous game.

Historically, a common approach to handling continuous action spaces has been to **discretize** them. Although this method can work well for smaller games, it inherently entails a loss in solution quality. For example, Kroer and Sandholm (2015) provided bounds on this quality degradation for extensive-form games. Furthermore, this approach fails to scale to high-dimensional observation and action spaces, because multidimensional environments cause a combinatorial explosion of points that scales exponentially with the number of dimensions. This scalability bottleneck is particularly intractable in highly complex spaces, such as generative adversarial networks (GANs) (Goodfellow et al., 2014), where a strategy is an entire probability distribution over images generated by a neural network. Additionally, explicit gradient information about the game is often unavailable in many practical applications. These limitations highlight a strong need for alternative approaches.

4.1.2 Double oracle

McMahan, Gordon, and Blum (2003) introduced the **double oracle** (DO) algorithm for normal-form games and proved its convergence. This algorithm maintains finite strategy sets for each player and iteratively expands them with best responses to an equilibrium of the induced finite sub-game. Adam et al. (2021) extended DO to two-player zero-sum continuous games, and Kroupa and Votroubek (2023) generalized it to n-player continuous games. This approach converged fast when the strategy space dimension was small and the generated subgames were not excessively large. For example, in two-player zero-sum cases, best responses were computed by selecting the optimal point of a uniform discretization for one-dimensional problems or by using a mixed-integer linear programming reformulation for Colonel Blotto games. However, this approach did not scale well to high-dimensional games with general payoffs where best-response computation was difficult. Moreover, estimating the finite subgame for stochastic games proved challenging and required many samples. Furthermore, this method failed to learn observation-dependent strategies that generalized across observations.

Li and Wellman (2021) extended the double oracle approach to n-player general-sum continuous Bayesian games by representing agents as neural networks optimized via NES (Wierstra et al., 2014). To approximate a pure-strategy equilibrium, they formulated the problem as a bi-level optimization and employed NES for both inner-loop best-response optimization and outer-loop regret minimization.

Double oracle algorithms generally maintained a set of static strategies that expanded on each iteration with approximate best responses to the opponents' meta-strategies. Several prominent algorithms built upon this foundational concept, including PSRO and its variants, which are described below.

Lanctot, Zambaldi, et al. (2017) introduced **policy-space response oracles** (PSRO), which generalized DO by defining the metagame's choices as continuous policies rather than discrete actions. PSRO successfully generalized fictitious self-play and allowed any meta-solver to compute new meta-strategies.

McAleer, Lanier, Fox, et al. (2020) introduced **pipeline PSRO** (P2SRO), a scalable method for finding approximate Nash equilibria (NE) in large zero-sum imperfect-information games. P2SRO parallelized PSRO with convergence guarantees by maintaining a hierarchical pipeline of reinforcement learning workers that trained against policies generated by lower hierarchy levels.

McAleer, Lanier, Wang, Baldi, and Fox (2021) introduced **extensive-form DO** (XDO) for two-player zero-sum games, which guaranteed linear convergence to an approximate NE relative to the number of infostates. Unlike PSRO, which mixed best responses at the root of the game, XDO mixed best responses at every infostate. The authors also proposed **neural XDO** (NXDO), which learned best responses through deep reinforcement learning and computed the metagame equilibrium using methods like NFSP (Heinrich and Silver, 2016) or DREAM (Steinberger, Lerer, and Brown, 2020). NXDO iteratively added reinforcement learning policies to a population while solving an extensive-form restricted game, a process that proved more efficient than solving matrix-form restricted games.

McAleer, Wang, et al. (2022) developed **anytime DO** (ADO), a tabular algorithm for two-player zero-sum games that guaranteed convergence to an NE while monotonically decreasing exploitability across iterations. They additionally introduced **anytime PSRO** (APSRO), an extension of ADO that calculated best responses using reinforcement learning.

McAleer, Lanier, Wang, Baldi, Sandholm, et al. (2024) created **self-play PSRO** (SP-PSRO), a method restricted to two-player zero-sum games that added approximately optimal stochastic policies to the population in each iteration. Instead of exclusively adding deterministic best responses to

the opponent's least exploitable population mixture, SP-PSRO learned and incorporated stochastic policies to avoid the need to enumerate all deterministic policies prior to convergence. However, SP-PSRO remained a normal-form algorithm that mixed at the root of the game tree, which McAleer, Lanier, Wang, Baldi, and Fox (2021) previously showed could require an exponential number of iterations to converge. The authors left the combination of SP-PSRO with extensive-form techniques like XDO and NXDO for future work.

Muller, Omidshafiei, et al. (2020) extended the theoretical underpinnings of PSRO by evaluating an alternative solution concept called α-Rank (Omidshafiei et al., 2019) instead of NE. They established convergence guarantees for this approach and identified formal links between NE and α-Rank.

Finally, Marris et al. (2021) proposed **joint policy-space response oracles** (JPSRO) for training agents in n-player, general-sum extensive-form games. This algorithm shifted the solution concept away from NE, utilizing CE and CCE instead (as described in Section 3.2).

4.1.3 Fictitious play

Brown (1951) introduced **fictitious play** (FP), a popular game-theoretic model of learning where players repeatedly play a game and choose a best response to their opponents' historical average strategies at each iteration. The average strategy profile converges to a Nash equilibrium (NE) in certain classes of games, including two-player zero-sum and potential games. Players update their beliefs simultaneously or alternately (Berger, 2007). Leslie and Collins (2006) introduced **generalized weakened FP** (GWFP), which generalizes FP by allowing approximate best responses and perturbed average strategy updates.

Perkins and Leslie (2014) extended stochastic FP to the continuous action space framework. They studied the limiting behavior of stochastic FP using the associated smooth best response dynamics on the space of finite signed measures. Using this approach, they showed that stochastic FP converges to an equilibrium in two-player zero-sum games.

Heinrich, Lanctot, and Silver (2015) introduced **full-width extensive-form FP** (XFP), which extends FP to extensive-form multi-step games. They also introduced **fictitious self-play** (FSP), a sample-based machine learning framework that implements GWFP in behavioral strategies. FSP avoids cycles by computing an approximate best response against a uniform mixture of all previous policies, which converges to an NE in two-player zero-sum games (Heinrich, Lanctot, and Silver, 2015). In FSP, players repeatedly play a game, store their experiences in memory, and mix between their best responses and average strategies to act cautiously. At each iteration, players replay their experience of play against opponents to compute an approximate best response, and similarly replay their own behavioral experience to learn a model of their average strategy.

Heinrich and Silver (2016) introduced **neural fictitious self-play** (NFSP), which combines FSP with neural network function approximation and deep reinforcement learning. An NFSP agent consists of two neural networks. The first network is trained via reinforcement learning from memorized experiences against fellow agents to learn an approximate best response to their historical behavior. The second network is trained via supervised learning from memorized experiences of the agent's own behavior.

Kamra, Gupta, Wang, et al. (2019) introduced **DeepFP**, an approximate FP algorithm for two-player games with continuous action spaces. They demonstrated stable convergence to equilibrium on several classic games and a large forest security domain. DeepFP represents players' approximate best responses via generative neural networks, which act as highly expressive implicit density

approximators. Because these implicit density models cannot be trained directly in the absence of gradients, the authors employed a game-model network to serve as a differentiable approximation of the players' utilities given their actions. This approach allows the networks to be trained end-to-end in a model-based learning regime.

Ganzfried (2021) introduced **redundant fictitious play** (RFP), an algorithm for approximating equilibria in continuous games, and applied it to a continuous Colonel Blotto game. Unlike our approach, this algorithm requires a best-response oracle as a subroutine (e.g., a mixed-integer linear program solver for the continuous Colonel Blotto game).

Vinyals et al. (2019) tackled StarCraft II, a real-time strategy game serving as a popular artificial intelligence benchmark, by introducing a multiagent reinforcement learning algorithm called **league training**. Noting that standard self-play algorithms learn rapidly but can chase cycles indefinitely, they extended FSP to compute best responses against a non-uniform mixture of opponents. Their league of potential opponents includes a diverse range of agents and their policies from both current and previous iterations. At each iteration, agents play games against opponents sampled from an agent-specific mixture policy, and their parameters are updated using an actor-critic reinforcement learning procedure based on those game outcomes. The league consists of three distinct types of agents that differ primarily in their mechanism for selecting the opponent mixture. First, main agents utilize a **prioritized fictitious self-play** (PFSP) mechanism that adapts mixture probabilities proportionally to each opponent's win rate against the agent, providing more opportunities to overcome problematic opponents. To recover the rapid learning of self-play, a main agent is selected as an opponent with a fixed probability. Second, main exploiter agents play exclusively against the current iteration of main agents to identify potential exploits and encourage the main agents to address their weaknesses. Third, league exploiter agents use a PFSP mechanism similar to the main agents, but are not targeted by main exploiters, aiming instead to find systemic weaknesses across the entire league. Both types of exploiter agents are periodically reinitialized to encourage diversity, allowing them to rapidly discover specialist strategies that are not necessarily robust against exploitation.

4.1.4 Evolution strategies

Bichler, Fichtl, Heidekrüger, et al. (2021) focused on symmetric auction models, assuming symmetric prior distributions and pure, symmetric equilibrium bidding strategies. To tackle these environments, they introduced a learning method called **neural pseudo-gradient ascent** (NPGA). NPGA represents strategies as neural networks and applies policy iteration based on gradient dynamics in self-play to provably learn local equilibria. The method follows the simultaneous gradient of the game and employs a smoothing technique to circumvent discontinuities in the *ex post* utility functions, which arise at the bid value where an arbitrarily small change makes the difference between winning and losing. They demonstrated that NPGA converges to a Bayesian Nash equilibrium across a wide variety of symmetric auction games.

Building on this work, Bichler, Kohring, and Heidekrüger (2023) extended the approach to asymmetric auctions, a setting that requires training multiple neural networks. They analyzed a wide variety of asymmetric auction models and showed that their method closely approximates Bayesian Nash equilibria in all environments where analytical solutions are known. Furthermore, they evaluated new, larger environments lacking analytical solutions and verified that the discovered solutions still approximate the equilibrium closely.

Li and Wellman (2021) tackled the problem of solving symmetric one-shot Bayesian games

characterized by high-dimensional type and action spaces, multiple players, general-sum payoffs, and no provided analytic structure. They represented agent strategies in parametric form as neural networks and applied NES to optimize them. For pure equilibrium computation, they formulated the problem as a bi-level optimization, using NES to implement both inner-loop best response optimization and outer-loop regret minimization. For mixed equilibrium computation, they adopted an incremental strategy generation framework where NES produces a finite sequence of approximate best-response strategies. Equilibria are then calculated over this finite strategy set via a model-based optimization process. Both methods use NES to search for strategies over the functional space of policies, relying solely on black-box simulation access to noisy payoff samples.

4.2 Prior work on infinite-player games

This section reviews existing literature on solving infinite-player games. Sandholm (2001) studied potential games with continuous player sets, such as random matching games with common payoffs and congestion games. Parise and Ozdaglar (2019) and Parise and Ozdaglar (2023) presented a framework for analyzing equilibria and designing interventions in large network games. These network games are modeled using a graphon, where a continuum of players receives payoffs dependent on their own action and a weighted average of others' actions. Building on this, graphon mean-field games (Caines and Huang, 2021; Cui and Koeppl, 2022) consider heterogeneous players with a continuous index space.

Perrin et al. (2020) analyzed continuous-time fictitious play on finite-state MFGs with common noise. They provided a convergence analysis and proved that the induced exploitability decreases at a linear rate. Muller, Rowland, et al. (2022) introduced mean-field PSRO, an adaptation of **policy-space response oracles** (PSRO) (Lanctot, Zambaldi, et al., 2017) that learns Nash, coarse-correlated, and correlated equilibria in MFGs. Pérolat et al. (2022) applied **online mirror descent** (OMD) to MFGs and proved that continuous-time OMD converges to a Nash equilibrium under certain conditions. Wang and Wellman (2023) solved MFGs using a double oracle algorithm by iteratively adding approximate best responses to the equilibrium of the empirical MFG formed by previously considered strategies. Wu et al. (2024) proposed a deep reinforcement learning algorithm that achieves population-dependent equilibria in MFGs without averaging or sampling from history. Zhang, Chen, and Di (2025) introduced **SemiSGD**, a stochastic semi-gradient descent method that treats a player's policy and the population distribution as a unified parameter to allow fully asynchronous, simultaneous updates. They also presented a **population-aware linear function approximation** (PA-LFA) for continuous state-action MFGs.

A parallel line of research applies reinforcement learning to MFGs under unknown dynamics. Guo et al. (2019) proposed a fitted Q-iteration scheme that provably converges to mean-field equilibria in discrete state-action MFGs by alternating between Q-value updates and a fixed-point update for the population distribution. Xie et al. (2021) extended this program to the finite-horizon setting, establishing sample-complexity bounds for provably efficient RL under contraction properties of the mean-field operator. Laurière et al. (2022) surveyed the field, cataloging both model-based and model-free approaches and highlighting open problems in non-stationary, non-contractive, and continuous state-action MFGs.

Comparison of MFGs to our setting: Classical MFG methods also address infinite-player games, but rely on the fundamental assumption that players are indistinguishable and individually negligible. Specifically, an individual's payoff depends only on its own action and the anonymous

mean-field density of the population.[1] In contrast, our approach does not require these symmetry or mean-field assumptions. Each player can have an arbitrary strategy space and a unique utility function that is not constrained to depend solely on aggregate behavior. Graphon MFGs (e.g., Caines and Huang (2021) and Cui and Koeppl (2022) relax indistinguishability by indexing players on a continuum, but still restrict each player's payoff to depend on others' actions only through a fixed graphon-weighted aggregate. Our framework admits arbitrary measurable utility functionals, including higher-order interactions and discontinuities, without a prescribed aggregation structure.

Agent-based modeling (ABM) is a bottom-up computational method that simulates autonomous, interacting agents, each with its own state and behavioral rules, within an environment to observe how system-level patterns emerge from local interactions (Epstein and Axtell, 1996; Bonabeau, 2002; Grimm et al., 2005; Macal and North, 2010). Our proposed framework acts as a differentiable analogue to ABM, leveraging the generalization capabilities of neural networks to accelerate learning.

4.3 Prior work on learning dynamics

In this section, we review prior work on gradient-based learning dynamics for games. They can be characterized by the ordinary differential equations (ODEs) shown in Table 4.1. Here, $\dot{x}$ is the time derivative of the current strategy profile, and $v = \operatorname{diag} \nabla u$ is the **simultaneous gradient**. Each component $v(x)_i = \nabla_i u(x)_i$ is the gradient of a player's utility with respect to their strategy. Additionally, $J = \nabla v$ is the Jacobian of the vector field v, $J^\top$ is its transpose, $J_a = \frac{1}{2}(J - J^\top)$ is its antisymmetric part, and J_o is its off-diagonal part (replacing its diagonal with zeroes). Dots indicate derivatives with respect to time, $\gamma > 0$ is a hyperparameter, and $v|_y$ denotes v evaluated at y rather than x. These methods were extensively analyzed in prior work, including Balduzzi et al. (2018), Letcher, Foerster, et al. (2019), Letcher, Balduzzi, et al. (2019), Mertikopoulos and Zhou (2019), Grnarova, Levy, et al. (2019), Mazumdar, Sastry, and Jordan (2025), Hsieh, Mertikopoulos, and Cevher (2021), and Willi et al. (2022).

The simultaneous gradient yields a vector field on the space of strategy profiles. Because this vector field may not be conservative (i.e., the gradient of a potential) as in gradient descent, standard gradient-based optimization methods often struggle, and trajectories can cycle around fixed points rather than converging to them. The actual optimization is done by discretizing each ODE in time. For example, SG is discretized as $x_{i+1} = x_i + \eta v_i$ and OP is discretized as $x_{i+1} = x_i + \eta v_i + \gamma(v_i - v_{i-1})$, where $\eta > 0$ is a stepsize and $v_j = v(x_j)$.

Simultaneous gradient ascent (SG) maximizes each player's utility independently, as if the other players are fixed. In the two-player zero-sum case, it is also known as **gradient descent ascent** (GDA). Goktas and Greenwald (2022b) studied min-max games with dependent strategy sets, where the strategy of the first player constrains the behavior of the second. They introduced two variants of GDA that assume access to a solution oracle for the optimal **Karush–Kuhn–Tucker** (KKT) multipliers of the games' constraints, proving a convergence guarantee.

Extragradient (EG) (Korpelevich, 1976) takes a step in the direction of the simultaneous gradient and uses the simultaneous gradient at that new point to take a step from the original point. Golowich et al. (2020) proved a tight last-iterate convergence guarantee for EG.

Optimistic gradient (OP) (Popov, 1980; Daskalakis et al., 2018; Hsieh, Iutzeler, et al., 2019) uses past gradients to predict future gradients and updates according to the latter.

[1] In anonymous games, identities do not matter, and only aggregate behavior affects outcomes.

Table 4.1: ODE corresponding to each method.

Method	$\dot{x}$
SG	v
EG	$v\|_{x+\gamma v}$
OP	$(I + \gamma \frac{\mathrm{d}}{\mathrm{d}t})v = v + \gamma \dot{v}$
CO	$(I - \gamma J^\top)v = v - \gamma \nabla \frac{1}{2}\|v\|^2$
SGA	$(I - \gamma J_a^\top)v$
LA	$(I + \gamma J_o)v$
SLA	$(I + \gamma J)v$
LOLA	$(I + \gamma J_o)v - \gamma \operatorname{diag} J_o^\top \nabla u$
LSS	$(I + J^\top J^{-1})v$
PCGD	$(I - \gamma J_o)^{-1}v$
ED	$-\nabla_x \sup_y \mathrm{NI}(x, y) = -\nabla_x \mathrm{Expl}(x)$
GNI	$-\nabla_x \mathrm{NI}(x, x + \gamma v)$

Consensus optimization (CO) (Mescheder, Nowozin, and Geiger, 2017) penalizes the magnitude of the simultaneous gradient, encouraging "consensus" between players that attracts them to fixed points.

Symplectic gradient adjustment (SGA) (Balduzzi et al., 2018), also known as Crossing-the-Curl (Gemp and Mahadevan, 2018), reduces the rotational component of game dynamics by using the antisymmetric part of the Jacobian.

Lookahead (LA) (Zhang and Lesser, 2010) excludes the diagonal components of the Jacobian. Each player predicts the behavior of other players after a step of naive learning, but assumes this step will occur independently of the current optimization.

In **symmetric lookahead** (SLA) (Letcher, 2018), instead of best-responding to opponents' learning, each player responds to *all* players learning, including themselves. It is a linearized version of EG (Domingo-Enrich, 2019, Lemma 1.35).

In **learning with opponent-learning awareness** (LOLA) (Foerster et al., 2018), a learner optimizes its utility assuming the opponent will take one naive learning step, rather than optimizing under the current parameters.

Mazumdar, Sastry, and Jordan (2025) proposed **local symplectic surgery** (LSS) to find local NE in two-player zero-sum games. It solves a linear system on each timestep, which is prohibitive for high-dimensional parameter spaces. Hence, the authors proposed a two-timescale approximation that updates the strategy profile while simultaneously improving an approximate solution to the linear system.

Competitive gradient descent (CGD) (Schäfer and Anandkumar, 2019) naturally generalizes gradient descent to the two-player setting. On each iteration, it jumps to the NE of a quadratically-regularized bilinear local approximation of the game. Its convergence and stability properties are robust to strong interactions between the players without adapting the stepsize.

Polymatrix competitive gradient descent (PCGD) (Ma et al., 2021) generalizes CGD to more than two players by jumping to the NE of a quadratically-regularized local polymatrix approximation of the game. The series expansion of PCGD to zeroth and first order in γ yields SG and LA (Willi et al., 2022, Proposition 4.4), respectively, since $(I - \gamma M)^{-1} = I + \gamma M + \gamma^2 M^2 + \dots$ for sufficiently small γ. Both CGD and PCGD require solving a linear system of equations on each

iteration, which is prohibitive for high-dimensional parameter spaces (Ma et al., 2021, p. 10).

Lockhart et al. (2019) introduced **exploitability descent** (ED), which directly minimizes exploitability against worst-case opponents to compute approximate equilibria in two-player zero-sum extensive-form games. They proved that when both players employ this optimization, the strategy profile asymptotically converges to an equilibrium, unlike extensive-form fictitious play (Heinrich, Lanctot, and Silver, 2015) and counterfactual regret minimization (Zinkevich et al., 2007), where convergence only pertains to time-average strategies. However, ED requires computing exact best responses y on each iteration, which is generally inefficient or intractable.

To address the intractability of exact best responses, Timbers et al. (2022) introduced approximate exploitability, which utilizes an approximate best response computed through search and reinforcement learning. This effectively generalizes the local best response metric used in poker (Lisý and Bowling, 2017).

Similarly, **gradient-based Nikaido–Isoda** (GNI) (Raghunathan, Cherian, and Jha, 2019) minimizes a local approximation of exploitability using local best responses $y = x + \gamma v$.

Goktas and Greenwald (2022a) recast the exploitability-minimization problem as a min-max optimization problem and obtained polynomial-time first-order methods for computing variational equilibria in convex-concave cumulative regret pseudo-games with jointly convex constraints. They presented two algorithms called **extragradient descent ascent** (EDA) and **augmented descent ascent** (ADA). We benchmark against EDA but not ADA because, unlike the other baselines, ADA requires *multiple* substeps of gradient ascent *per timestep* to approximate a best response, making its accuracy highly dependent on the quality of that inner search.

Fiez, Chasnov, and Ratliff (2019) investigated the convergence of learning dynamics in Stackelberg games with continuous action spaces, characterizing conditions under which attracting critical points of simultaneous gradient ascent are Stackelberg equilibria in zero-sum games. They developed a gradient-based update for the leader, paired with a follower employing either a best response strategy or a gradient-play update rule, yielding algorithms that provably converge to a Stackelberg equilibrium given appropriate initialization.

Fiez, Chasnov, and Ratliff (2020) studied learning in Stackelberg games, establishing connections between Nash and Stackelberg equilibria alongside the limit points of simultaneous gradient ascent. They designed gradient-based learning dynamics that emulate the natural structure of a Stackelberg game using the implicit function theorem, providing convergence analysis for deterministic and stochastic updates. However, our focus remains on finding *Nash* equilibria rather than Stackelberg equilibria.

Fiez, Jin, et al. (2022) considered minimax optimization $\min_x \max_y f(x, y)$ in the context of two-player zero-sum games, where the min-player minimizes f under the assumption that the max-player will subsequently maximize it. In their framework, the min-player plays against **smooth algorithms** deployed by the max-player rather than exact full maximization, which is generally NP-hard. Their algorithm guarantees monotonic progress, avoids limit cycles, and finds an appropriate stationary point in a polynomial number of iterations.

Wellman, Tuyls, and Greenwald (2025) surveyed **empirical game-theoretic analysis** (EGTA), which uses simulation to generate data from which one can induce an empirical game model. The machine learning approach to empirical game modeling is a form of regression in which the input is a set of (profile, payoff-vector) pairs and the output is the vector of empirical utility functions. These techniques can be used to infer a complete empirical game model from an incomplete one.

In a **generative adversarial network** (GAN) (Goodfellow et al., 2014), a generator learns to produce fake data while a discriminator learns to distinguish it from real data. To stabilize GANs

and address mode collapse, Metz et al. (2017) introduced a method defining the generator objective with respect to an unrolled optimization of the discriminator. Grnarova, Levy, et al. (2019) proposed using an approximation of the game-theoretic **duality gap** as a performance measure for GANs, and later proposed using this measure as the training objective itself to provide convergence guarantees (Grnarova, Kilcher, et al., 2021).

Gemp, Savani, et al. (2022) proposed **average deviation incentive descent with adaptive sampling** (ADIDAS), which iteratively improves an approximation to a NE through joint play by tracing a homotopy path defining a continuum of regularized equilibria. To encourage iterates to remain near this path, the algorithm minimizes the average deviation incentive via stochastic gradient descent.

Mazumdar, Ratliff, and Sastry (2020) analyzed the limiting behavior of competitive gradient-based learning algorithms using dynamical systems theory. They characterized a non-negligible subset of local NE that are actively avoided when agents employ gradient-based learning algorithms.

Mertikopoulos and Zhou (2019) examined the convergence of no-regret learning in continuous games, focusing on "dual averaging," where players take small steps along their individual utility gradients and mirror the output back to their action sets. They introduced the notion of variational stability, showing that stable equilibria are locally attracting with high probability and globally stable equilibria are globally attracting with probability 1.

Mertikopoulos and Staudigl (2018) extended this analysis to noisy gradient input, establishing almost-sure convergence of continuous-time stochastic gradient flows to variationally stable equilibria under diminishing step sizes. This setting has a connection to our use of zeroth-order pseudo-gradient estimators (Sections 2.8, 5.1, and 5.3.1), where gradient information is replaced by noisy finite-difference estimates. Bravo, Leslie, and Mertikopoulos (2018) studied N-player concave games under bandit feedback, in which each player observes only its own scalar utility rather than a gradient. They proved no-regret bounds and convergence to Nash equilibria under appropriate monotonicity conditions. This feedback model closely matches the setting of our JPSPG estimator (Section 5.3.1), which uses only utility evaluations. Conversely, Vlatakis-Gkaragkounis et al. (2020) established a negative result: in general finite games, no-regret learning dynamics need not converge to mixed Nash equilibria, even in the time-average sense. This motivates our use of an exploitability-based objective and learned best responses (Section 5.2) rather than reliance on time-average no-regret guarantees.

While most previous work on minimax optimization focused on classical notions of equilibria from simultaneous games, Jin, Netrapalli, and Jordan (2020) proposed a mathematical definition of local optimality in sequential game settings, which include GANs and adversarial training. Due to the nonconvex-nonconcave nature of these problems, minimax is generally not equal to maximin, making the order in which players act crucial.

To address the rotational behavior of ordinary gradient dynamics in these sequential settings, Wang, Zhang, and Ba (2020) proposed **follow-the-ridge** (FR) for two-player zero-sum sequential games, an algorithm that provably converges exclusively to local minimax points.

Tsaknakis and Hong (2021) proposed an algorithm for finding the fractional Nash equilibria (FNEs) of a two-player zero-sum game with non-convex local cost functions and access only to local stochastic gradients. Their approach reformulates the problem as minimizing the **regularized Nikaido–Isoda** (RNI) function. Unlike our work, their method is restricted to two-player zero-sum games and relies on nontrivial subroutines, assuming subproblems are solved to a given accuracy using external methods like projected gradient descent.

Willi et al. (2022) showed that the original formulation of LOLA is inconsistent because it models

other agents as naive learners rather than LOLA agents, a flaw previously suggested as a cause for LOLA's failure to preserve stable fixed points. They formalized consistency and demonstrated that **higher-order LOLA** (HOLA) solves this inconsistency if it converges. They also proposed **consistent LOLA** (COLA), a method that relies on no more than second-order derivatives to learn consistent update functions under mutual opponent shaping, functioning successfully even when HOLA fails to converge.

Perolat et al. (2022) introduced **DeepNash**, an autonomous agent that mastered the imperfect-information game Stratego from scratch via self-play using a model-free deep reinforcement learning method. Its key component, the **Regularised Nash dynamics** (R-NaD) algorithm, modifies the underlying multiagent learning dynamics to converge to an approximate NE instead of cycling around it.

Similarly, Qin et al. (2022) proposed PORL, a no-regret style reinforcement learning algorithm for continuous action tasks with a proven last-iterate convergence guarantee.

Bao and Zhang (2022) proposed **double follow-the-ridge** (double-FTR), an algorithm with local convergence guarantees to differential NE in general-sum two-player differential games.

Chapter 5

Completed work

In this section, we describe the work we have already completed toward our ultimate goal. Before doing so, we provide a brief summary of each contribution.

1. In Martin and Sandholm (2023), we proposed **randomized policy networks**, a method to compute approximate Nash equilibria in continuous-action games without access to gradient information, by combining zeroth-order optimization with randomized policy networks that model observation-dependent mixed strategies, showing it can efficiently find high-quality approximate equilibria even when pure-strategy equilibria do not exist and gradients are not available.

2. In Martin and Sandholm (2025b), we proposed **ApproxED: Approximate exploitability descent via learned best responses**. It consists of two methods that minimize an approximation of exploitability for continuous-action games by training the strategy profile jointly with either learned best-response functions or ensembles of candidate best responses, enabling direct descent on an exploitability proxy. Experiments on continuous games (and GAN training) show these methods find lower-exploitability strategies and outperform prior baselines.

3. In Martin and Sandholm (2025d), we proposed **joint-perturbation simultaneous pseudo-gradient** (JPSPG), a zeroth-order method for finding approximate Nash equilibria in games with continuous strategy spaces and no gradient access that performs a single joint perturbation across all players so the number of utility evaluations per iteration is constant (not linear) in the number of players. The paper shows this yields large wall-time improvements on many-player problems (including auctions and other settings with expensive or discontinuous payoffs) and empirically outperforms per-player perturbation baselines.

4. In Martin and Sandholm (2025e), we proposed **player-to-strategy networks** (P2SN), a neural-network representation that maps player identities to strategies so a single model can represent strategy profiles across countably or uncountably infinite players by leveraging function approximation. We also proposed **shared-parameter simultaneous gradient** (SPSG), an algorithm that generalizes simultaneous gradient ascent to train P2SNs toward approximate Nash equilibria, and show it converges on a variety of infinite-player games (including games with infinitely many states/actions and discontinuous utilities).

In each of the following subsections, we describe each contribution in greater detail.

5.1 Finding mixed-strategy equilibria of continuous-action games without gradients using randomized policy networks

To our knowledge, Martin and Sandholm (2023) was the first work to solve general continuous-action games with unrestricted mixed strategies and without any gradient information. In that work, we studied the problem of computing an approximate Nash equilibrium of a continuous-action game without access to gradients. Such game access is common in reinforcement learning settings, where the environment is typically treated as a black box. To tackle this problem, we applied zeroth-order optimization techniques that combine smoothed gradient estimators with equilibrium-finding dynamics. We modeled players' strategies using artificial neural networks. In particular, we used randomized policy networks to model mixed strategies. These take noise in addition to an observation as input and can flexibly represent arbitrary observation-dependent, continuous-action distributions. Being able to model such mixed strategies is crucial for tackling continuous-action games that lack pure-strategy equilibria. We evaluated the performance of our method using an approximation of the Nash convergence metric from game theory, which measures how much players can benefit from unilaterally changing their strategy. We applied our method to continuous Colonel Blotto games, single-item and multi-item auctions, and a visibility game. The experiments showed that our method can quickly find a high-quality approximate equilibrium. Furthermore, they showed that the dimensionality of the input noise is crucial for performance. In particular, noise of too low dimension (or no noise, which yields a deterministic policy) results in high exploitability.

5.1.1 Method

Bichler, Fichtl, Heidekrüger, et al. (2021) modeled strategies using neural networks, where each player's policy network takes as input a player's observation and outputs an action. As described in Section 4.1.4, these policy networks were then trained using NPGA, which uses Gaussian smoothing and applies simultaneous gradient ascent. Their policy networks can only model pure strategies, since the output action is deterministic with respect to the input observation.

We also model strategies using neural networks, with one crucial difference: our policy network f_θ takes as input the player's observation *o together with noise z* from some **fixed latent distribution**, such as the standard multivariate Gaussian distribution. Thus the output $a = f_\theta(o, z)$ of the network is *random* given o. The network can then learn to transform this randomness into a desired action distribution. This lets us model mixed strategies, which is especially desirable in games that lack pure-strategy equilibria. Some approaches in the literature use the output of a policy network to parameterize some parametric distribution on the action space, such as a Gaussian mixture model. However, taking the randomness *as input* and letting the neural network *reshape* it as desired allows us to model arbitrary distributions more flexibly.

Figure 5.1 illustrates the high-level structure of a randomized policy network. It takes as input an observation and random noise, concatenates them, passes the result through a feedforward neural network, and outputs an action. The **dimensionality of noise** fed into a randomized policy network is an important hyperparameter. In Section B.2, we review the literature that studies the relation between input noise dimension and representational power in neural network-based generative models.

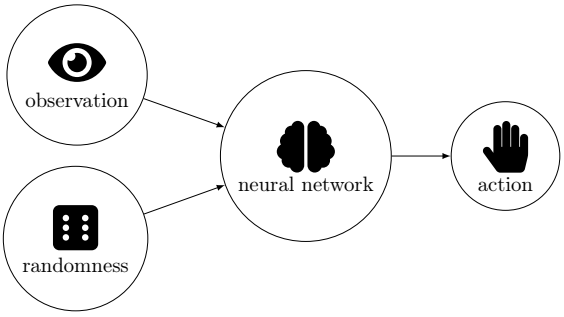


Figure 5.1: High-level structure of a randomized policy network.

Like Bichler, Fichtl, Heidekrüger, et al. (2021), we train these networks through NPGA, which can be performed in a massively parallelized or distributed fashion, as described in Algorithm 1.

On any iteration, there is a set of available workers $\mathcal{J}$. Each worker is assigned the task of computing a pseudo-gradient for a particular player. The vector $\{a_j\}_{j\in\mathcal{J}}$ contains the assignment of a player for each worker. Each worker's **pseudorandom number generator (PRNG)** is initialized with the same fixed seed. On each iteration, each worker evaluates the utility function (generally the most expensive operation and bottleneck for training) twice to compute the finite difference required for the pseudo-gradient. It then sends this computed finite difference (a single scalar) to the other workers. In turn, it receives the other workers' scalars. (This is called an "allgather" operation in parallel computing.) Therefore, the information that needs to be passed between workers is minimal. This greatly reduces the required cross-worker bandwidth compared to schemes that pass parameters or gradients between workers, which can be prohibitively expensive for large models.

This massively parallelizes Algorithm 1 of Bichler, Fichtl, Heidekrüger, et al. (2021) ("NPGA using ES gradients"). Simultaneously, it generalizes Algorithm 2 of Salimans et al. (2017) ("Parallelized Evolution Strategies"), which also uses shared seeds, to the multiplayer setting, with separate gradient evaluations and optimizers for each player. Furthermore, it allows for the possibility of setting the worker-player assignments a_j and perturbation noise scales ε_j dynamically over time, provided that this is done consistently across workers (for example, based on their common state variables). Vanilla gradient descent, momentum gradient descent, optimistic gradient descent, or other optimization algorithms can be used.

The set of available workers can also change dynamically over time. If a worker leaves or joins the pool, the coordinator notifies all workers of its ID so they can remove it from, or add it to, their $\mathcal{J}$ sets. The new worker is brought up to speed by passing it the current PRNG state, strategy profile parameters, and optimizer states (what state information is needed depends on the algorithm used, for example, whether momentum is used).

5.1.2 Experiments

We tested our approach on various benchmark games from the literature, evaluating the exploitability of the resulting learned strategy profile. To initialize our networks, we use He initialization (He et al., 2015), which is widely used for feedforward networks with ReLU-like activation functions. It initializes bias vectors to zero and weight matrices with normally-distributed entries scaled by $\sqrt{2/n}$, where n is the layer's input dimension. We use the ELU activation function (Clevert, Unterthiner,

Algorithm 1 Distributed multiagent pseudo-gradient ascent

Input: $\mathcal{I}$ is the set of players, u is the utility function
  initialize PRNG state with fixed seed
  $\mathbf{x} \leftarrow$ initial strategy profile
  for $i \in \mathcal{I}$ **do**
    $S_i \leftarrow \text{init}(\mathbf{x}_i)$ ▷ initial state of optimizer i
  loop
    $\mathcal{J} \leftarrow$ set of available workers
    for $j \in \mathcal{J}$ **do**
      $a_j \in \mathcal{I}$ ▷ set a player; can be set dynamically
      $\varepsilon_j \in \mathbb{R}_{>0}$ ▷ set a scale; can be set dynamically
      $\mathbf{z}_j \sim N(\mathbf{0}, \mathbf{I}_{\dim \mathbf{x}_i})$ where $i = a_j$
    $j \leftarrow$ own worker ID
    $\delta_j \leftarrow \frac{u(\mathbf{x}[i \mapsto \mathbf{x}_i + \varepsilon_j \mathbf{z}_j])_i - u(\mathbf{x}[i \mapsto \mathbf{x}_i - \varepsilon_j \mathbf{z}_j])_i}{2\varepsilon_j}$ where $i = a_j$
    send δ_j to coordinator, receive δ from coordinator
    for $i \in \mathcal{I}$ **do**
      $\mathcal{K} \leftarrow \{j \in \mathcal{J} \mid a_j = i\}$ ▷ workers assigned i
      $\mathbf{v}_i \leftarrow \frac{1}{|\mathcal{K}|} \sum_{j \in \mathcal{K}} \delta_j \mathbf{z}_j$ ▷ i's pseudo-gradient
      $S_i, \mathbf{x}_i \leftarrow \text{step}(S_i, \mathbf{v}_i)$ ▷ step optimizer i

and Hochreiter, 2016) for hidden layers. Like Bichler, Fichtl, Heidekrüger, et al. (2021), we use 2 hidden layers with 10 neurons each.

We illustrate the performance of our method by plotting the exploitability across optimization steps. Noise dimensions 0–4 are shown in blue, orange, green, red, and purple respectively. Each of these is run for 20 trials. Solid lines indicate means across trials. Bands indicate a confidence interval for this mean with a confidence level of 0.95. These confidence intervals are computed using bootstrapping (Efron, 1979), specifically the bias-corrected and accelerated (BCa) method (Efron, 1987).

For the gradient estimator, we use the Gaussian distribution with scale $\sigma = 10^{-2}$, $N = 1$ perturbation vectors, and the central-difference stencil (2 evaluations per step). For the optimizer, we use standard simultaneous gradient ascent with a learning rate of 10^{-6}. To estimate exploitability (see the BR computation described in Section 3.3), we use 100 observation samples and 300 state samples (given each observation). We use a 100-point discretization of the action space for the auctions and visibility game. For the continuous Colonel Blotto games, we use a 231-point discretization of the action space. It is obtained by enumerating all partitions of the integer 20 into 3 parts and renormalizing them to sum to 1.

5.1.2.1 Continuous Colonel Blotto

The original **Colonel Blotto game** was introduced by Borel (1921). It is a two-player zero-sum game in which two players distribute resources over several battlefields. A battlefield is won by whoever devotes the most resources to it. A player's payoff is the number of battlefields they win. This game models many real-world situations of conflict or competition that involve resource allocation, such as political campaigns, research and development, national security, and systems defense.

Various variants have been studied in the literature. Gross and Wagner (1950) analyzed a **continuous** variant in which both players have continuous, possibly unequal budgets. They obtained exact solutions for various special cases, including all 2-battlefield cases and all 3-battlefield cases with equal budgets. Washburn (2013) generalized to the case where battlefield values are unequal across battlefields. Kovenock and Roberson (2021) generalized to the case where battlefield values are also unequal across players. Adamo and Matros (2009) studied a variant in which players have incomplete information about the other player's resource budgets. Kovenock and Roberson (2011) studied a model where the players are subject to incomplete information about the battlefield valuations. Boix-Adserà, Edelman, and Jayanti (2021) analyzed the natural multiplayer generalization of the continuous Colonel Blotto game.

The general case with continuous allocations, heterogeneous budgets, heterogeneous battlefield values across both players and battlefields, and several players is as follows. Suppose there are J battlefields. Let b_i be the budget of player i. Let v_{ij} be the value to player i of battlefield j. Player i's action space is the standard J-simplex dilated by their budget: $A_i = \{a_{ij} \in \mathbb{R} \mid a_{ij} \geq 0, \sum_j a_{ij} = b_i\}$. Player i's reward function is $r_i(a) = \sum_j v_{ij} w_{ij}(a)$ where w_{ij} is the probability that i wins j. Ties are broken uniformly at random.

We test on the 2-player, 3-item continuous Colonel Blotto game with fixed homogeneous budgets and valuations. This game was studied by Gross and Wagner (1950), who derived an exact solution. They give the following geometric description of the equilibrium strategy: "[The player] inscribes a circle within [the triangle] and erects a hemisphere upon this circle. He next chooses a point from a density uniformly distributed over the surface of the hemisphere and projects this point straight down into the plane of the triangle... He then divides his forces in respective proportion to the triangular areas subtended by [this point] and the sides." The exact solution is illustrated in Figure 5.2.

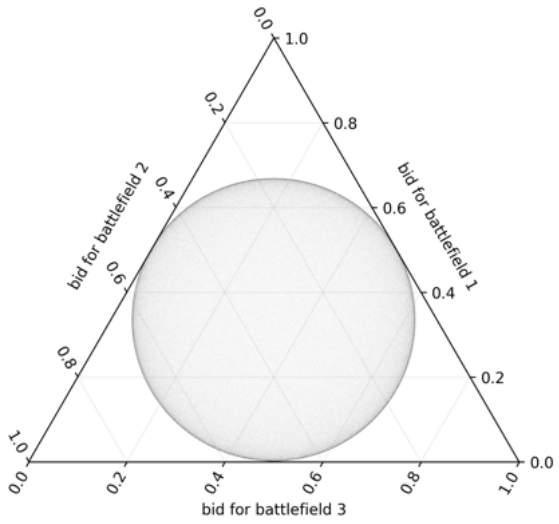


Figure 5.2: Exact solution for 2-player, 3-item continuous Colonel Blotto, shown as a 2D histogram of samples. Darker means higher density.

Actions in the continuous Colonel Blotto game are points on the standard simplex. Thus we use

a softmax activation function for the output layer of the randomized policy network. We describe an efficient way to compute best responses in Section B.3.

Figure 5.3 illustrates the performance of our method on the continuous Colonel Blotto game with 2 players and 3 battlefields. Since the game has no pure-strategy Nash equilibrium, deterministic strategies perform badly, as expected. 1-dimensional noise results in slightly better performance, but does not let players randomize well enough to approximate the equilibrium. On the other hand, noise of dimension 2 and higher is sufficient for good performance. The very slight increase in exploitability after 10^8 steps is most likely due to fluctuations introduced by the many sources of stochasticity in the training process, including the game and gradient estimates, as well as the fact that we are training a multi-layer neural network. Even in the supervised learning setting, loss does not always decrease monotonically.

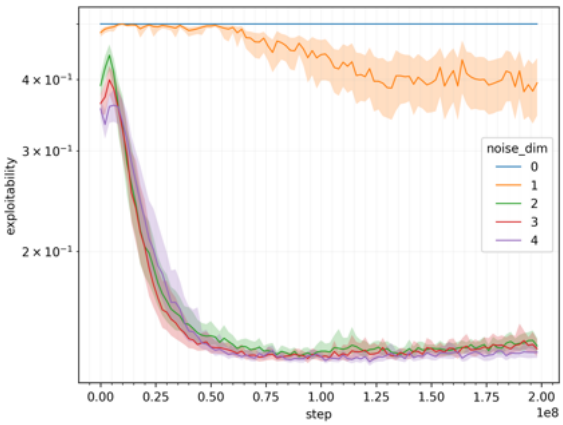


Figure 5.3: Exploitabilities for 2-player, 3-item continuous Colonel Blotto.

Figure 5.4 illustrates the strategies at different stages of training for one trial that uses 2-dimensional noise. Each scatter plot is made by sampling 10^5 actions from each player's strategy.

Figure 5.5 also illustrates performances on the continuous Colonel Blotto game with 2 players and 3 battlefields. This time, however, the budgets for each player are sampled from the standard uniform distribution and revealed to both players. Thus each player must adjust their action distribution accordingly. To our knowledge, prior approaches (Adam et al., 2021; Kroupa and Votroubek, 2023; Ganzfried, 2021) did not learn strategies that can generalize across different parameters (like budgets and valuations), which requires the use of function approximators such as neural networks.

5.1.2.2 Complete-information auction

An auction is a mechanism by which a set of **items** are sold to a set of **bidders**, who have valuations for items or sets thereof. Auctions play a central role in the study of markets and are used in a wide range of real-world contexts (Krishna, 2002), such as advertising, commodities, radio spectrum allocation, real estate, and more. To evaluate their method, Bichler, Fichtl, Heidekrüger, et al. (2021) used auctions as a benchmark.

In a **single-item sealed bid auction**, bidders simultaneously submit bids and the highest bidder wins the item. Let $w_i(a)$ be the probability i wins given action profile a, where ties are broken

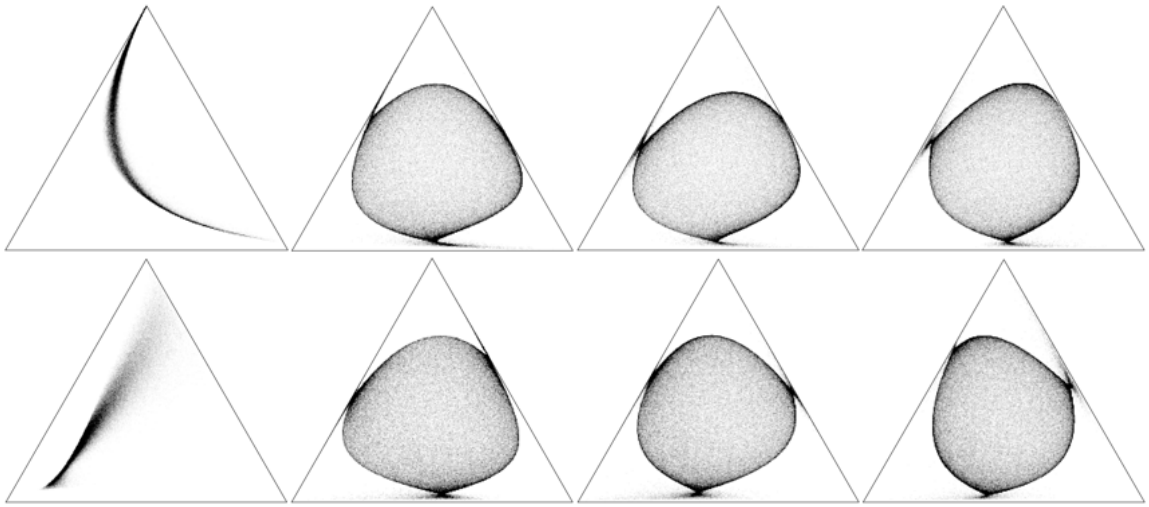


Figure 5.4: Learned strategies for 2-player, 3-item continuous Colonel Blotto, illustrated as 2D histograms of action samples. Top to bottom: Players 1 and 2. Left to right: Across training. Darker means higher density. Each histogram uses 10^4 samples.

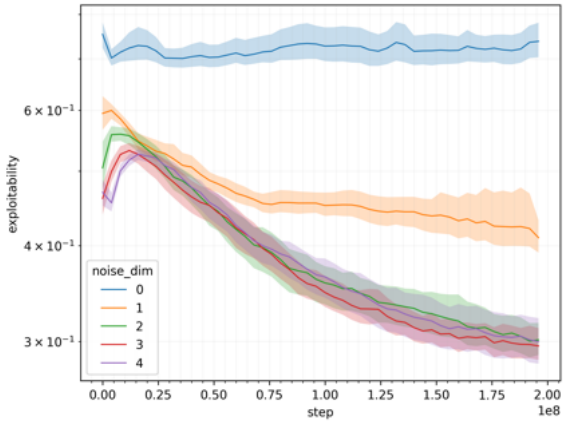


Figure 5.5: Exploitabilities for 2-player, 3-item continuous Colonel Blotto (with random budgets).

uniformly at random. Let $v_i(\omega)$ be the item's value for the ith player given state ω. In a kth-price **winner-pay** auction, the winner pays the kth highest bid: $r_i(\omega, a) = w_i(a)(v_i(\omega) - a_{(k)})$, where $a_{(k)}$ is the kth highest bid. In an **all-pay** auction, each player always pays their bid: $r_i(\omega, a) = w_i(a)v_i(\omega) - a_i$. More details about each type of auction can be found in Section B.1.

The all-pay complete-information auction is widely used to model lobbying for rents in regulated and trade protected industries, technological competition and R&D races, political campaigns, job promotions, and other contests (Baye, Kovenock, and Vries, 1996). It lacks pure-strategy equilibria

(Baye, Kovenock, and Vries, 1996). We test our method on this auction with 2 players.

This game was studied by Baye, Kovenock, and Vries (1996), who derived an exact solution. They show that with homogeneous valuations ($v_1 = v_2 = \ldots = v_n$) there exists a unique symmetric equilibrium and a continuum of asymmetric equilibria. All of these equilibria are payoff equivalent, as is the expected sum of the bids (revenue to the auctioneer). In the symmetric equilibrium, each player randomizes uniformly on $[0, v_1]$. The exact solution is illustrated in Figure 5.6.

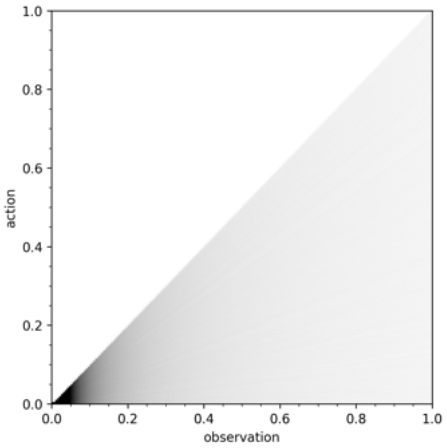


Figure 5.6: Exact solution for the 2-player complete-information auction.

To ensure the output is non-negative, we use a squaring function in the output layer, rather than a ReLU function like Bichler, Fichtl, Heidekrüger, et al. (2021). The reason is that, as we found in our experiments, ReLU can easily cause degenerate initializations: if the randomly-initialized neural network happens to map all of the unit interval (the observation space) to negative bids, no gradient signal can be received and the network is stuck.

Figure 5.7 and Figure 5.8 illustrate performances and strategies for the complete-information all-pay auction. Recall that these auctions have no pure-strategy equilibria. Thus, as expected, deterministic strategies perform poorly. As with Colonel Blotto games, our experiments in these auction settings show that the ability to flexibly model mixed strategies is crucial for computing approximate Nash equilibria in certain auction settings.

5.1.2.3 Asymmetric-information auction

The 2-player 1st-price winner-pay asymmetric-information auction (in which bidder 1 is perfectly informed of the common value of the object, and bidder 2 is completely uninformed) also lacks pure-strategy equilibria (Krishna, 2002, Section 8.3). In particular, the second player (who is uninformed) must randomize its bid.

This game was studied by Krishna (2002, Section 8.3), who derived an exact solution. Bidder 1 bids according to the strategy $\beta(v) = \mathrm{E}[V \mid V \leq v]$. In our case, $V \sim \mathcal{U}([0, 1])$, so $\beta(v) = \frac{v}{2}$. Bidder 2 chooses a bid at random from the interval $[0, \mathrm{E}[V]]$ according to the distribution defined by $H(b) = \mathrm{P}[\beta(V) \leq b]$. In our case, this reduces to the distribution $\mathcal{U}([0, \frac{1}{2}])$. The exact solution is illustrated in Figure 5.9.

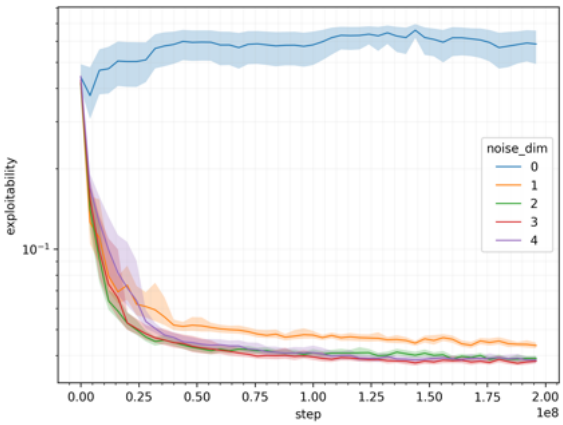


Figure 5.7: Exploitabilities for the 2-player all-pay complete-information auction.

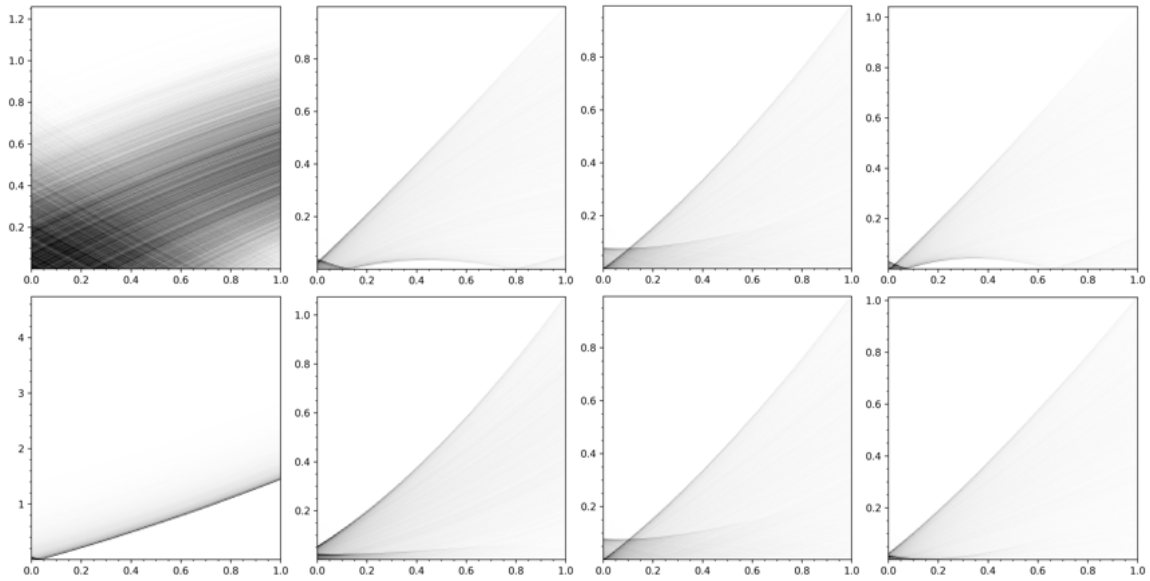


Figure 5.8: Learned strategies for the 2-player all-pay complete-information auction, illustrated as 1D histograms of action samples. Top to bottom: Players 1 and 2. Left to right: Across training. Darker means higher density. X and Y axes denote observation and bid, respectively. Each histogram uses 10^4 samples per observation.

Figure 5.10 illustrates performances for the asymmetric information auction.

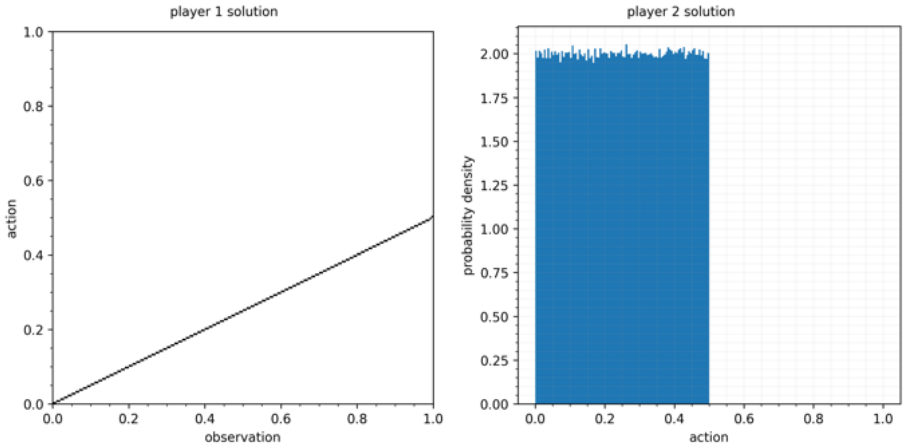


Figure 5.9: Exact solution for the asymmetric-information auction. Left to right: Players 1 and 2.

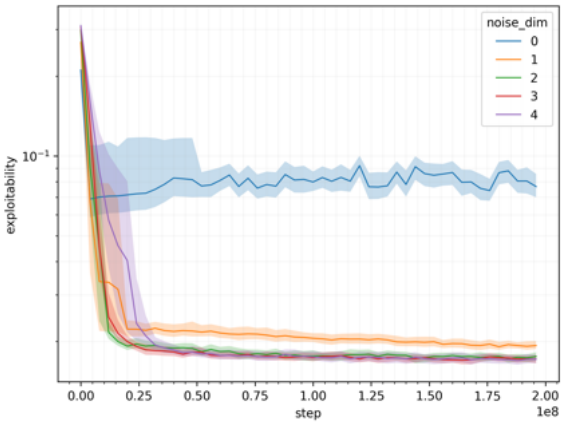


Figure 5.10: Exploitabilities for the asymmetric-information auction.

5.1.2.4 Chopstick auction

Multi-item auctions are of great importance in practice, for example in strategic sourcing (Sandholm, 2013) and radio spectrum allocation (Milgrom and Segal, 2014; Milgrom and Segal, 2020). However, deriving equilibrium bidding strategies for multi-item auctions is notoriously elusive. A rare notable instance where equilibrium strategies have been derived is the **chopstick auction** (Szentes and Rosenthal, 2003b; Szentes and Rosenthal, 2003a). In this auction, 3 chopsticks are sold simultaneously in separate first-price sealed-bid auctions. There are 2 bidders, and it is common knowledge that a pair of chopsticks is worth \$1, a single chopstick is worth nothing by itself, and 3 chopsticks are worth the same as 2. Here, pure strategies are triples of non-negative real numbers (bids).

This game was studied by Szentes and Rosenthal (2003b) and Szentes and Rosenthal (2003a), who derived an exact solution. They describe it as follows: "The supports of the mixtures that

generate the symmetric equilibria in both the first- and second-price cases, turn out to be the surfaces of regular tetrahedra, and the distributions themselves turn out to be uniform on these surfaces. In addition, in each case all the points inside the tetrahedron are pure best responses to the equilibrium mixture." Mathematically, define the tetrahedron T as the convex hull of the four points $(\frac{1}{2}, \frac{1}{2}, 0)$, $(\frac{1}{2}, 0, \frac{1}{2})$, $(0, \frac{1}{2}, \frac{1}{2})$, and $(0, 0, 0)$. Then the uniform probability measure on the 2-dimensional surface of T generates a symmetric equilibrium. Furthermore, all points inside the tetrahedron are pure best responses to this equilibrium mixture. The exact solution is illustrated in Figure 5.11.

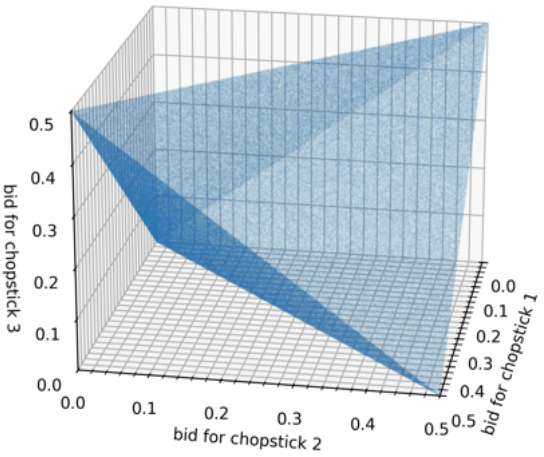


Figure 5.11: Exact solution for the chopstick auction, shown as a 3D scatterplot of samples.

We benchmark on the chopstick auction since it is a rare case of a multi-item auction with a known analytic equilibrium, so we can compare our output to an exact equilibrium. It is also a canonical case of simultaneous separate auctions under combinatorial preferences.

Figure 5.12 and Figure 5.13 illustrate performances and strategies for the chopstick auction. The latter figure shows that, with more epochs, the strategies better approximate a tetrahedron, which is the analytic equilibrium (as discussed below).

Here we encounter an interesting phenomenon. Recall that this game has a symmetric equilibrium generated by the uniform measure on the surface of a tetrahedron. Although the tetrahedron itself is 3-dimensional, its surface is only 2-dimensional. Thus one may wonder whether 2-dimensional noise is sufficient, that is, whether the network can learn to project this lower-dimensional manifold out into the third dimension while "folding" it in the way required to obtain the surface of the tetrahedron. Through our experiments, we observe that 2-dimensional noise suffices to (approximately) match the performance of higher-dimensional noise. Thus the **intrinsic dimension** of the equilibrium action distribution (as opposed to the extrinsic dimension of the ambient space in which it is embedded) seems to be the decisive factor.

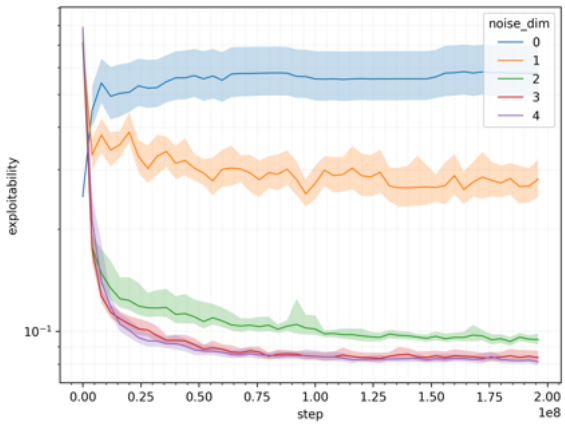


Figure 5.12: Exploitabilities for the chopstick auction.

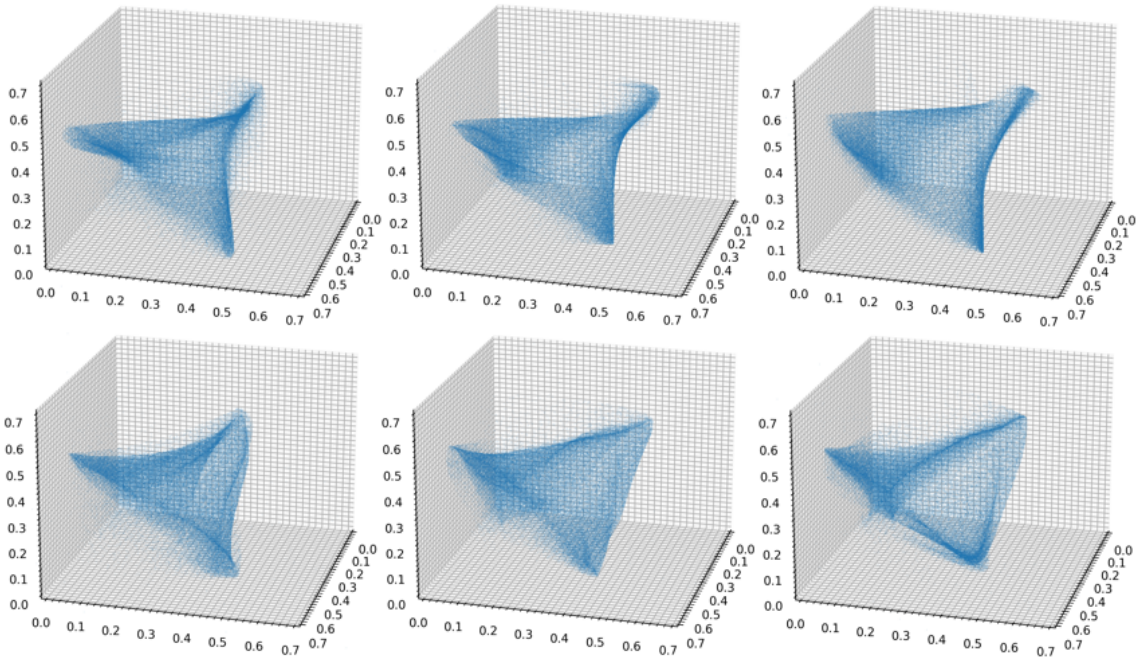


Figure 5.13: Learned strategies for the 2-player, 3-item chopstick auction, shown as 3D scatterplots of action samples. Top to bottom: Players 1 and 2. Left to right: Across training. X, Y, and Z axes denote bid for each item. Each histogram uses 10^5 samples.

5.1.2.5 Visibility game

Lotker, Patt-Shamir, and Tuttle (2008) introduced the *visibility game*, a noncooperative, complete-information strategic game. In this game, each player i chooses a point $x_i \in [0, 1]$. Their payoff is

the distance to the next higher point, or to 1 if x_i is the highest. This game models a situation where players seek to maximize their *visibility time*, and is a variant of the family of "timing games" (Fudenberg and Tirole, 1991).

It resembles the "war of attrition" game formalized by Smith (1974). In this game, both players are engaged in a costly competition and they need to choose a time to concede. More formally, the first player to concede (called "leader") gets a smaller payoff than the other player (called "follower"). Furthermore, the payoff to the leader strictly decreases as time progresses. That is, conceding early is better than conceding late.

Lotker, Patt-Shamir, and Tuttle (2008) proved that the n-player visibility game has no pure equilibrium, but has a unique mixed equilibrium, which is symmetric. In the 2-player case, up to a set of measure zero, there is a unique equilibrium whose strategies have probability densities $p(x) = 1/(1-x)$ when $0 \leq x \leq 1 - 1/\mathrm{e} \approx 0.632$ and 0 otherwise. Under this equilibrium, each player's expected payoff is $1/\mathrm{e}$. The exact solution is illustrated in Figure 5.14.

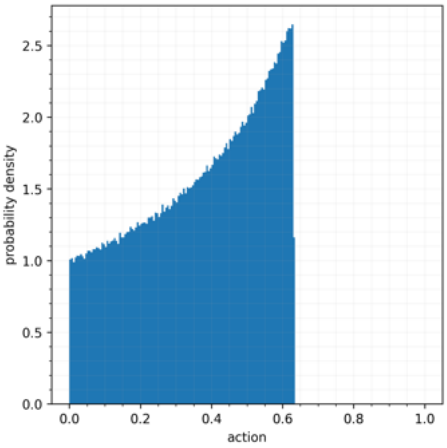


Figure 5.14: Exact solution for the 2-player visibility game.

Figure 5.15 illustrates performances on the 2-player visibility game. Figure 5.16 illustrates strategies during training for a trial with 1-dimensional noise. The players' action distributions converge to the expected distribution, including the distinctive cutoff at ≈ 0.632.

As expected, 0-dimensional noise, which yields deterministic strategies, performs very poorly. More interestingly, there is a noticeable gap in performance between 1-dimensional noise, which matches the dimensionality of the action space, and higher-dimensional noise. That is, using noise of higher dimension than the action space accelerates convergence in this game.

The experiments showed that our method can quickly compute a high-quality approximate equilibrium for these games. Furthermore, they showed that the dimensionality of the input noise is crucial for representing and converging to equilibrium. In particular, noise of too low dimension (or no noise, which yields a deterministic policy) results in failure to converge. Randomized policy networks flexibly model observation-dependent action distributions. Thus, in contrast to prior work, we can flexibly model mixed strategies and directly optimize them in a "black-box" game with access only to payoffs.

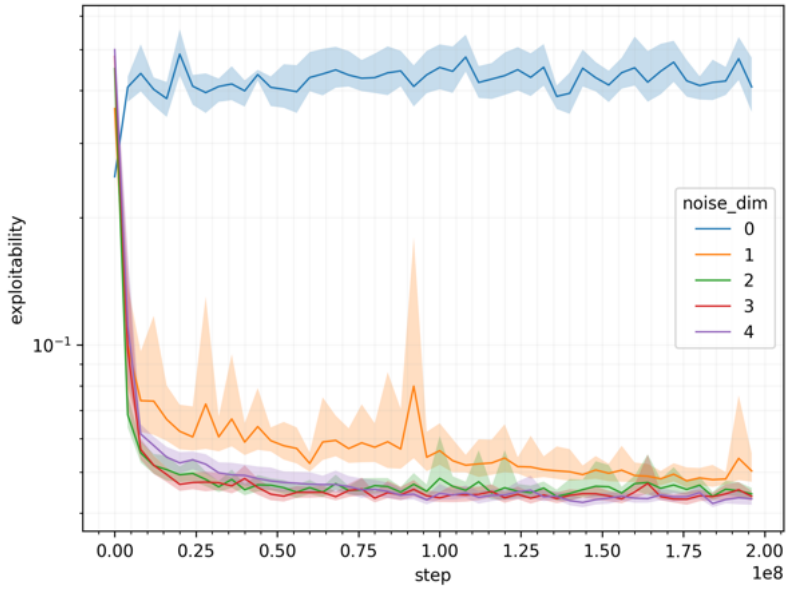


Figure 5.15: Exploitabilities for the 2-player visibility game.

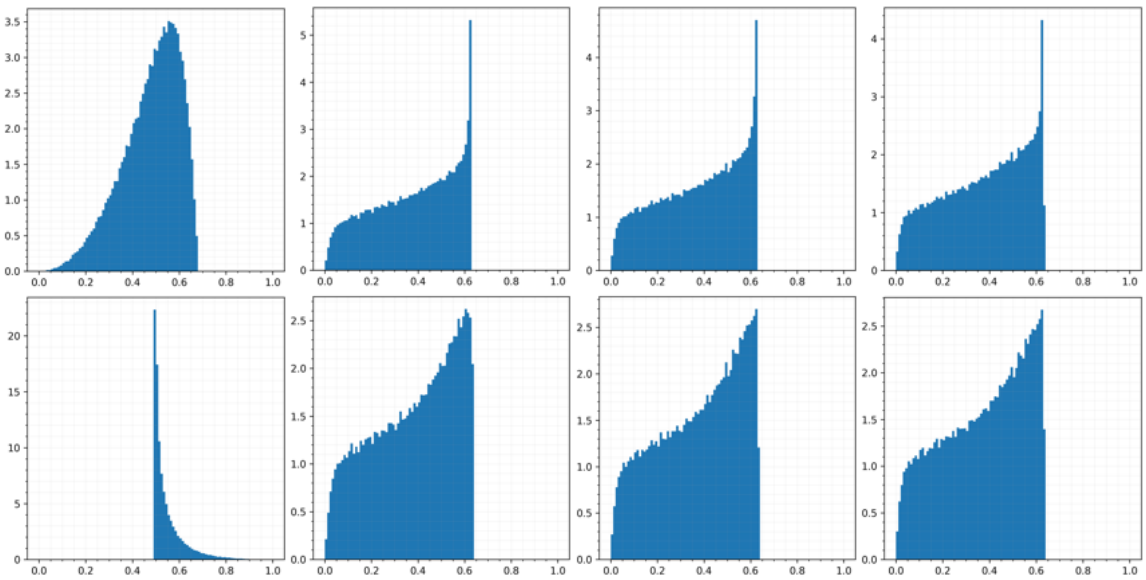


Figure 5.16: Learned strategies for the 2-player visibility game, shown as 1D histograms. Top to bottom: Players 1 and 2. Left to right: Across training. X and Y axes denote the action and probability density, respectively. Each histogram uses 10^5 samples.

5.2 ApproxED: Approximate exploitability descent via learned best responses

In Martin and Sandholm (2025b), we study the problem of finding an approximate Nash equilibrium of games with continuous action sets. The standard measure of closeness to Nash equilibrium is exploitability, which measures how much players can benefit from unilaterally changing their strategy. We propose two new methods that minimize an approximation of exploitability with respect to the strategy profile. The first method uses a learned best-response function, which takes the current strategy profile as input and outputs candidate best responses for each player. The strategy profile and best-response functions are trained simultaneously, with the former trying to minimize exploitability while the latter tries to maximize it. The second method maintains an ensemble of candidate best responses for each player. In each iteration, the best-performing elements of each ensemble are used to update the current strategy profile. The strategy profile and ensembles are simultaneously trained to minimize and maximize the approximate exploitability, respectively. We evaluate our methods on various continuous games and GAN training, showing that they outperform prior methods.

5.2.1 Method

We are given a utility function u[1] and our goal is to find an NE. Since the exploitability $\text{Expl}(x) = \sup_y \text{NI}(x, y)$ is non-negative everywhere, and zero exactly at NE, we reformulate the problem of finding an NE (if one exists) as *finding a global minimum of the exploitability function*. That is, we wish to solve the min-max optimization problem $\inf_x \sup_y \text{NI}(x, y)$. This is equivalent to finding a minimally-exploitable strategy for a two-player zero-sum meta-game with utility function NI. To find a minimum, we could try performing gradient descent on NI, like ED does. However, ED requires best-response oracles. In general games, exact best responses can be difficult or intractable to compute. Thus we need alternative solutions.

To solve this problem, we can try to perform gradient descent on x and ascent on y simultaneously: $\dot{x} = -\nabla_x \text{NI}(x, y)$, $\dot{y} = \nabla_y \text{NI}(x, y)$. Unfortunately, this approach can fail even in simple games. For example, consider the simple bilinear game with $u(x, y) = xy$. The unique Nash equilibrium is at the origin. However, simultaneous gradient ascent fails to converge to it, and instead cycles around it indefinitely. The essence of this cycling problem is that Player 2 has to "relearn" a good response to Player 1 every time Player 1's strategy switches sign. This is a general problem for games. "Small" changes in other players' strategies can cause "large" (discontinuous) changes in a player's best response. When such changes occur, players have to "relearn" how to respond to the other players' strategies. We propose two methods to tackle this problem. These are described in the next two subsections, respectively.

5.2.1.1 Best-response functions

For this method, we conceptually reformulate the problem as

$$\operatorname*{argmin}_{x \in \mathcal{X}} \sup_{y \in \mathcal{Y}} \text{NI}(x, y) = \operatorname*{argmin}_{x \in \mathcal{X}} \text{NI}(x, b(x)) \tag{5.1}$$

[1] For exposition, we assume utility functions are differentiable. If they are not, we can replace gradients with *pseudo-gradients*, as described in Section 2.8.

where $b \in \mathcal{X} \to \mathcal{Y}$ is a function that satisfies

$$b(x) \in \operatorname*{argmax}_{y \in \mathcal{Y}} \mathrm{NI}(x, y) \tag{5.2}$$

Since b is a *function*, it can map different strategies for Player 1 to different strategies for Player 2. Thus it can, at least in principle, *immediately* adapt to Player 1's strategy x, without forgetting prior solutions, and thus avoid the cycling problem.

More precisely, suppose $\mathcal{Y}$ is compact and $\mathrm{NI} \in \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is continuous in its second argument. Let $x \in \mathcal{X}$. By the extreme value theorem, a continuous real-valued function on a non-empty compact set attains its extrema. Therefore, there exists $y \in \mathcal{Y}$ such that $\mathrm{NI}(x, y) = \sup_{y \in \mathcal{Y}} \mathrm{NI}(x, y)$. Since this is true for every $x \in \mathcal{X}$, there exists a function $b \in \mathcal{X} \to \mathcal{Y}$ such that, for every $x \in \mathcal{X}$, $\mathrm{NI}(x, b(x)) = \sup_{y \in \mathcal{Y}} \mathrm{NI}(x, y)$. In other words, b is a **best-response function**. Even when $\mathcal{Y}$ is not compact and NI does not attain its extrema, one can define a best-response *value* for any $x \in \mathcal{X}$ as $\sup_{y \in \mathcal{Y}} \mathrm{NI}(x, y)$, provided the latter exists. In that case, we have the following. Let $\varepsilon > 0$ and $x \in \mathcal{X}$. Any function gets arbitrarily close to its supremum (continuity is not required). Therefore, there exists a $y \in \mathcal{Y}$ such that $\mathrm{NI}(x, y) + \varepsilon \geq \sup_{y \in \mathcal{Y}} \mathrm{NI}(x, y)$. Therefore, there exists a function $b \in \mathcal{X} \to \mathcal{Y}$ such that, for every $x \in \mathcal{X}$, $\mathrm{NI}(x, b(x)) + \varepsilon \geq \sup_{y \in \mathcal{Y}} \mathrm{NI}(x, y)$. That is b is an ε-approximate best-response function.

To find x and b simultaneously, we can perform simultaneous gradient ascent:

$$\dot{x} = -\nabla_x \mathrm{NI}(x, b(x)) \tag{5.3}$$

$$\dot{b} = +\nabla_b \mathrm{NI}(x, b(x)) \tag{5.4}$$

where $\nabla_x \mathrm{NI}(x, b(x))$ is a total (not partial) derivative. That is, the best response function tries to *increase* the exploitability while the strategy profile tries to *decrease* it. Since b is a function, Player 1's changing behavior poses no fundamental hindrance to it learning good responses and "saving" them for later use if Player 1's behavior changes. It could even learn a good approximation to the true best-response function, leaving Player 1 to face a simple standard optimization problem.

If $\mathcal{X}$ is infinite and $\mathcal{Y}$ is nontrivial, $\mathcal{X} \to \mathcal{Y}$ has infinite dimension. To represent and optimize b in practice, we need a finite-dimensional parameterization of (a subset of) this function space. In particular, if b is parameterized by θ and is (approximately) surjective onto $\mathcal{Y}$, then $\mathrm{Expl}(x) = \sup_{y \in \mathcal{Y}} \mathrm{NI}(x, y) \approx \sup_{\theta \in \Theta} \mathrm{NI}(x, b_\theta(x))$. Inspired by this idea, we propose jointly optimizing x and θ according to the following ODE system.

$$\dot{x} = -\nabla_x \mathrm{NI}(x, b_\theta(x)) \tag{5.5}$$

$$\dot{\theta} = +\nabla_\theta \mathrm{NI}(x, b_\theta(x)) \tag{5.6}$$

We call our method **approximate exploitability descent with learned best-response functions** (ApproxED-BRF). Its structure is depicted in Figure 5.17. As the figure illustrates, it maps a *strategy profile* to a profile of *best responses* for each player to the other players. It is an all-to-all function. We emphasize that this cross-player propagation of information occurs only for the purpose of *finding approximate best responses* in order to train *the strategy parameters*, and does not mean, for example, that players now get to observe each other's actions within the real game itself.

The best-response function can take on many possible forms. One possibility is to use a neural network. These have the advantages described in Section 1.1. They are also, by far, the most popular function approximators used in game solving. Therefore, we use this approach in our experiments.

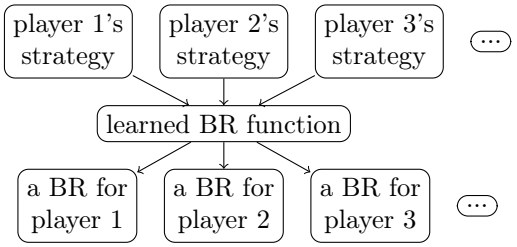


Figure 5.17: Structure of the BRF, in the case of 3 players.

On the other hand, we also experiment with settings where each player's strategy is *itself* represented by a neural network. In this case, the best-response functions take those networks' parameters as input. In other words, we adapt the concept of **hypernetworks** (Schmidhuber, 1992; Ha, Dai, and Le, 2017; Lorraine and Duvenaud, 2018; MacKay et al., 2019; Bae and Grosse, 2020) to the game-theoretic context. We note that, to obtain good performance, the learned best response function does not need to represent the true best response function (which may be discontinuous) *exactly*, but only *approximately*. Our experimental results indicate that the approximation yielded by the neural network performs well across a wide class of games.

5.2.1.2 Best-response ensembles

For this method, we conceptually reformulate the problem as

$$\operatorname*{argmin}_{x\in\mathcal{X}} \sup_{y\in\mathcal{Y}} \mathrm{NI}(x,y) \approx \operatorname*{argmin}_{x\in\mathcal{X}} \max_{j\in\mathcal{J}} \mathrm{NI}(x,y_j) \tag{5.7}$$

where $\mathcal{J}$ is a finite set of indices, and $x \in \mathcal{X}$ and $y \in \mathcal{J} \to \mathcal{Y}$ are trainable parameters. In other words, we use an **ensemble** of $|\mathcal{J}|$ responses to x, where the *best* response is selected automatically by evaluating x against each y_j and taking the one that attains the best value. Each individual y_j is a *strategy* for the original game. Since there are multiple responses in the ensemble, each one can "focus on" tackling a particular "type" of behavior from x without having to change drastically when the latter changes. We can then train x and y simultaneously: $\dot{x} = -\nabla_x \max_{j\in\mathcal{J}} \mathrm{NI}(x,y_j)$, $\dot{y} = \nabla_y \max_{j\in\mathcal{J}} \mathrm{NI}(x,y_j)$. That is, x improves against the best y_j, while the best y_j improves against x. Ties are broken in lexicographic order (smaller indices first). This allows for symmetry breaking if the ensemble elements are initially equal and the game is deterministic.

There is an issue with the aforementioned scheme, however. If one of the ensemble elements y_j strictly dominates the others for all encountered x, then the other elements will never be selected under the maximum operator. Thus they will never have a chance to change, improve performance, and thus contribute. In that case, the scheme degenerates to ordinary simultaneous gradient ascent. We observed this degeneracy in some games.

To solve this issue, we introduced the following approach. To give *all* y_j some chance to improve, while incentivizing them to "focus" on particular types of x rather than cover all cases, we use a **rank-based weighting** approach. Specifically, we let $\dot{y} = \nabla_y \operatorname{mix}_{j\in\mathcal{J}} \mathrm{NI}(x,y_j)$ where $\operatorname{mix}_{j\in\mathcal{J}} a_j = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} r_j a_j$ and $r_j \in \{1,\dots,|\mathcal{J}|\}$ is the ordinal rank of element j. This makes better elements receive a higher weight. Thus the best y_j has the most incentive to adapt against the

current x, while others have less incentive, but still some nonetheless.[2]

Our method is defined by the following ODE system.

$$\dot{x} = -\nabla_x \sum_{i \in \mathcal{I}} \left(\max_{j \in \mathcal{J}} u(x[i \mapsto y_{ij}])_i - u(x)_i \right) \tag{5.8}$$

$$\dot{y} = +\nabla_y \sum_{i \in \mathcal{I}} \left(\operatorname*{mix}_{j \in \mathcal{J}} u(x[i \mapsto y_{ij}])_i - u(x)_i \right) \tag{5.9}$$

Here, $y = \{y_{ij}\}_{i \in \mathcal{I}, j \in \mathcal{J}}$ is an ensemble of $|\mathcal{J}|$ responses for each individual player $i \in \mathcal{I}$. We call our method **approximate exploitability descent with learned best-response ensembles** (ApproxED-BRE). Its structure is depicted in Figure 5.18.

One can compute the $u(x[i \mapsto y_{ij}])_i$ and their gradients in parallel for $i \in \mathcal{I}, j \in \mathcal{J}$. Hence, with access to $O(|\mathcal{I}||\mathcal{J}|)$ cores, the method can run approximately as fast as standard simultaneous gradient ascent. The ensemble size can be as big as the amount of memory and number of cores or workers available allows. If a practitioner has access to parallel computing infrastructure, they can scale up the ensemble size as much as possible, until all of the available parallelism is used up, thus reaping the benefits of more coverage of the response space while incurring little to no penalty in wall time. An interesting direction for future research would be to analyze the effect of the ensemble size.[3]

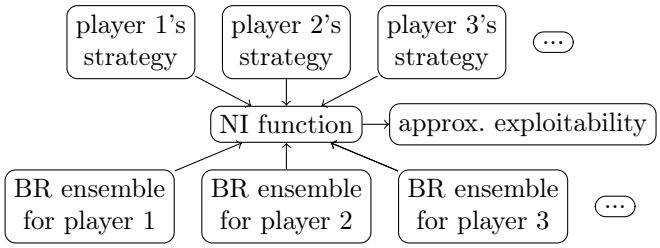


Figure 5.18: Structure of the BRE, in the case of 3 players.

5.2.2 Experiments

In this section, we present some of our experiments. The full list of experiments can be found in Martin and Sandholm (2025b). We use a learning rate of $\eta = 10^{-3}$ and $\gamma = 10^{-1}$. For BRF's best-response function, we use a fully-connected network with a hidden layer of size 32, the tanh activation function, and Glorot weight initialization (Glorot and Bengio, 2010). We do not try to find the best neural architecture, because this problem comprises an entire field, may be task-specific, and is not the focus of our paper. Thus our experiments are conservative, in the sense that our technique could perform even better compared to the baselines if engineering effort were spent tuning the neural network. For BRE, we use ensembles of size 10 for each player. For each experiment, we ran

[2]Furthermore, since the weight of each ensemble element depends only on the rank or order of values, it is invariant under monotone transformations of the utility function.

[3]For clarity, we emphasize that each individual strategy (including each individual element of an ensemble) could, in general, represent a *mixed* strategy of the original game. Thus the support size of the NE (in terms of *pure strategies* of the original game) is not necessarily directly related to the ensemble size.

64 trials. In our plots, solid lines show the mean across trials, and bands show its standard error. For games with stochastic utility functions, we used a batch size of 64.

5.2.2.1 Saddle-point game

For illustration, we start with a simple game. This is a two-player zero-sum game with actions that are real numbers and utility function $u_1(x, y) = -u_2(x, y) = xy$. It has a unique NE at the origin.

Our experimental results are shown in Figure 5.19. Our methods converge fastest. Our methods also take a more direct path to the equilibrium.

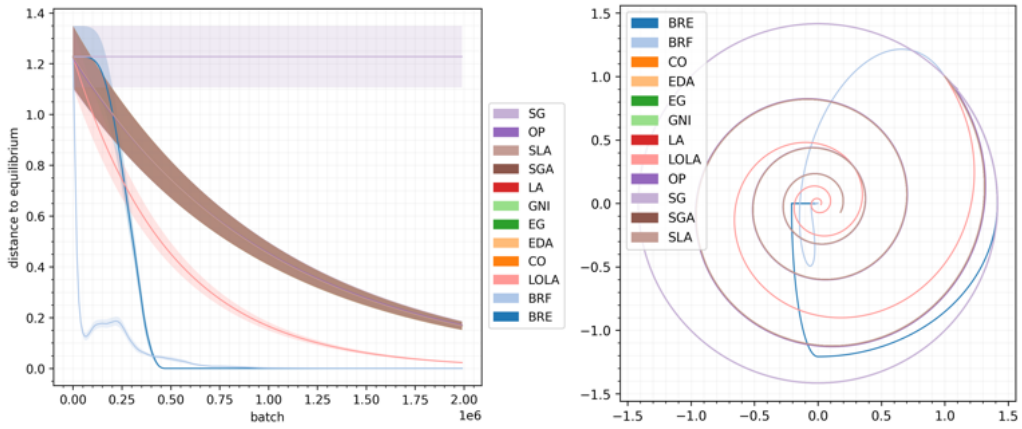


Figure 5.19: Saddle-point game. Left: Distances to equilibrium. Right: Trajectories.

5.2.2.2 Generalized matching pennies

Matching pennies (MP) is a classic two-player zero-sum normal-form game with 2 actions per player. It can be generalized to n players (Jordan, 1993; Leslie and Collins, 2003) by letting $u(a)_i = [\![(a_i = a_{i+1 \bmod n}) \oplus (i = n-1)]\!]$, where $\oplus$ denotes exclusive disjunction. That is, each player seeks to match the next, but the last player seeks to *un*match the first. Utilities are shown in Table 5.1.

	H	T
H	$+1$	-1
T	-1	$+1$

Table 5.1: Utilities for matching pennies (2 players), from the perspective of player 1.

This game has a unique mixed-strategy NE where each player mixes uniformly over its actions. The 3-player game's NE is locally unstable in a strong sense (Jordan, 1993). More precisely, discrete-time fictitious play (Brown, 1951) fails to converge, and instead enters a limit cycle asymptotically.

Our experimental results are shown in Figure 5.20. In the 2-player game, our methods converge fastest. In the 3-player game, our methods converge, while the rest diverge.

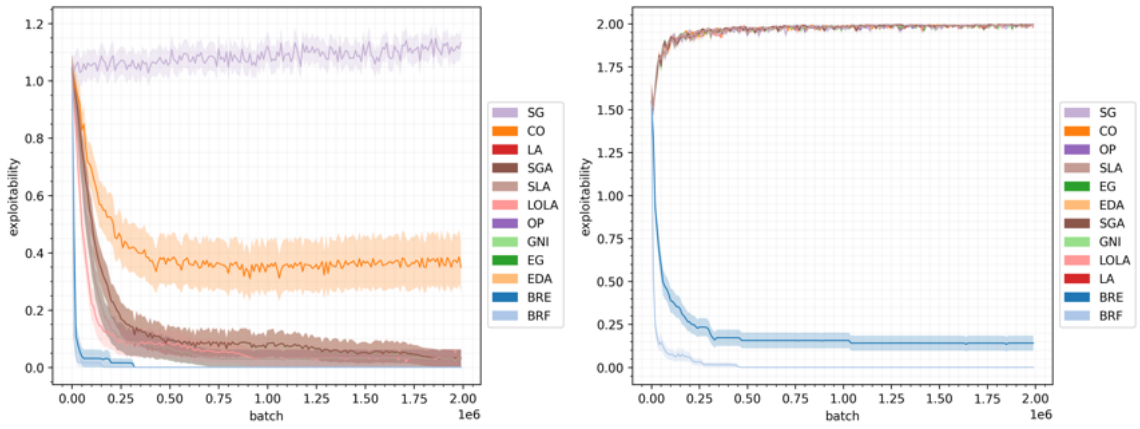


Figure 5.20: Matching pennies with 2 and 3 players.

5.2.2.3 Generalized rock paper scissors

Rock paper scissors (RPS) is a classic two-player zero-sum normal-form game with 3 actions per player. It has a unique mixed-strategy NE where each player mixes uniformly over its actions. Cloud, Wang, and Kerr (2023, p. 7) generalize RPS to n actions by letting $u_1(a) = -u_2(a) = [\![a_2 - a_1 = 1 \mod n]\!] - [\![a_1 - a_2 = 1 \mod n]\!]$. Utilities are shown in Table 5.2 and Table 5.3.

	R	P	S
R	0	-1	$+1$
P	$+1$	0	-1
S	-1	$+1$	0

Table 5.2: Utilities for rock paper scissors (3 actions), from the perspective of player 1.

	A	B	C	D
A	0	-1	0	$+1$
B	$+1$	0	-1	0
C	0	$+1$	0	-1
D	-1	0	$+1$	0

Table 5.3: Utilities for rock paper scissors (4 actions), from the perspective of player 1.

Our experimental results are shown in Figure 5.21. Our methods converge fastest.

5.2.2.4 Shapley game

This is a two-player normal-form game with 3 actions per player. Utilities are shown in Table 5.4.

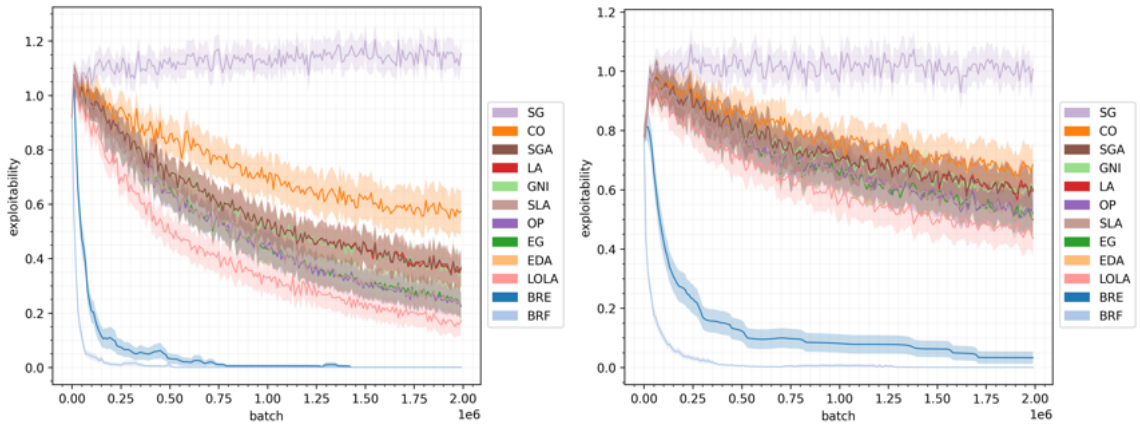


Figure 5.21: Rock paper scissors with 3 and 4 actions.

	A	B	C
A	1	0	0
B	0	1	0
C	0	0	1

	A	B	C
A	0	1	0
B	0	0	1
C	1	0	0

Table 5.4: Utilities for the Shapley game (players 1 and 2).

This game was introduced by Shapley (1964, p. 26), and is a classic example of a game for which fictitious play (Brown, 1951; Berger, 2007) diverges. (Instead, fictitious play cycles through the cells with 1's in them, with ever-increasing lengths of play in each of these cells.)

Our experimental results are shown in Figure 5.22. Our methods converge, while the rest diverge and perform almost identically to each other.

5.2.2.5 Glicksberg–Gross game

This is a two-player zero-sum normal-form game with continuous action sets $\mathcal{A}_i = [0, 1]$ and utility function $u_1(x, y) = -u_2(x, y) = \frac{(1+x)(1+y)(1-xy)}{(1+xy)^2}$. Glicksberg and Gross (1953) analyzed this game and proved that it has a unique mixed-strategy NE where each player's strategy has a cumulative distribution function of $F(t) = \frac{4}{\pi} \arctan \sqrt{t}$. To model mixed strategies, we use the following implicit density model. We feed a sample from a 1-dimensional standard normal distribution into a fully-connected network with one hidden layer of size 32 and output layer of size 1. The output is squeezed to the unit interval using the logistic sigmoid function.

Our experimental results are shown in Figure 5.23. Our methods converge fastest.

5.2.2.6 Continuous security game

Security games are used to model defender-adversary interactions in many domains, such as the protection of infrastructure like airports, ports, and flights (Kamra, Gupta, Fang, et al., 2018), as

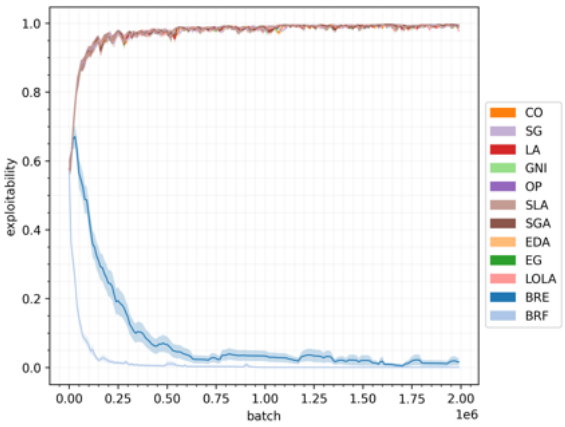


Figure 5.22: Shapley game.

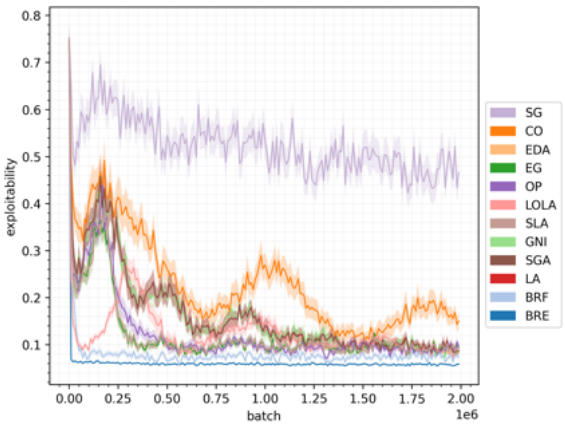


Figure 5.23: Glicksberg–Gross game.

well as wildlife, fisheries, and forests (Kar et al., 2017; Sinha et al., 2018). Security games are often modeled with Stackelberg equilibrium as the solution concept, which coincides with NE in zero-sum security games and certain structured general-sum games (Korzhyk et al., 2011). Many security games have continuous action spaces. These have been studied by Kamra, Fang, et al. (2017), Kamra, Gupta, Fang, et al. (2018), and Kamra, Gupta, Wang, et al. (2019). Consider the following game. Let $\mathcal{S} = [0, 1]^2$. The attacker chooses a point $x \in \mathcal{S}$. Simultaneously, the defender chooses n points $y_i \in \mathcal{S}$. Let $d = \inf_{i \in [n]} \|x - y_i\|$ be the distance between the attacker's point and the defender's closest point. The defender receives a utility of $\exp(-d^2)$, and the attacker receives $-\exp(-d^2)$. Thus the defender seeks to be close to the attacker, while the opposite is true for the attacker.

Our experimental results are shown in Figure 5.24. Our methods perform best.

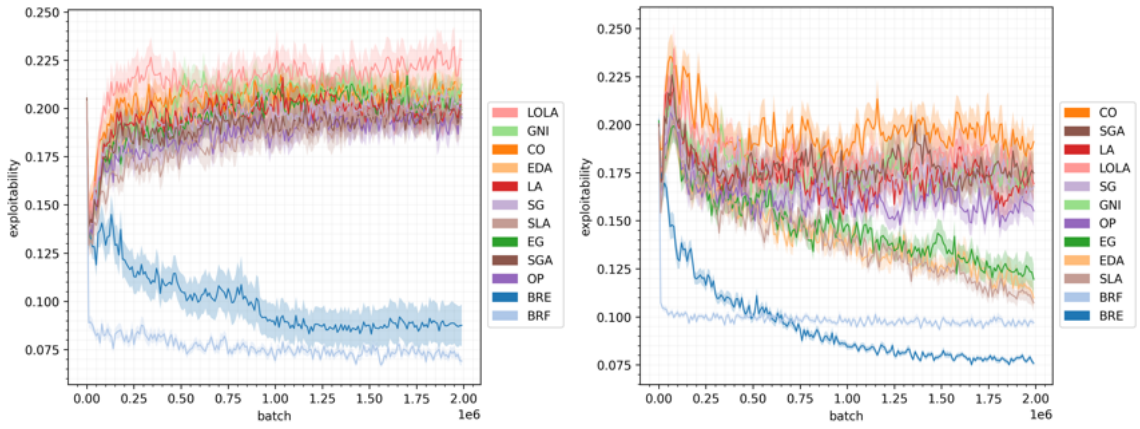


Figure 5.24: Security game with 1 and 2 points.

5.2.2.7 Poker games

Kuhn poker is a variant of poker introduced by Kuhn (1950). It is a two-player zero-sum imperfect-information game. A 3-player variant was introduced by Szafron, Gibson, and Sturtevant (2013), and was one of the largest three-player games to be solved analytically to date. 2-player Kuhn poker has a 12-dimensional strategy space per player (24 in total). 3-player Kuhn poker has a 32-dimensional strategy space per player (96 in total). Thus the utility function for these games is high-dimensional and nonlinear, making them a good benchmark. We use the poker implementation of OpenSpiel (Lanctot, Lockhart, et al., 2019).

Our experimental results are shown in Figure 5.25. Our methods converge fastest. The rest perform almost identically to each other.

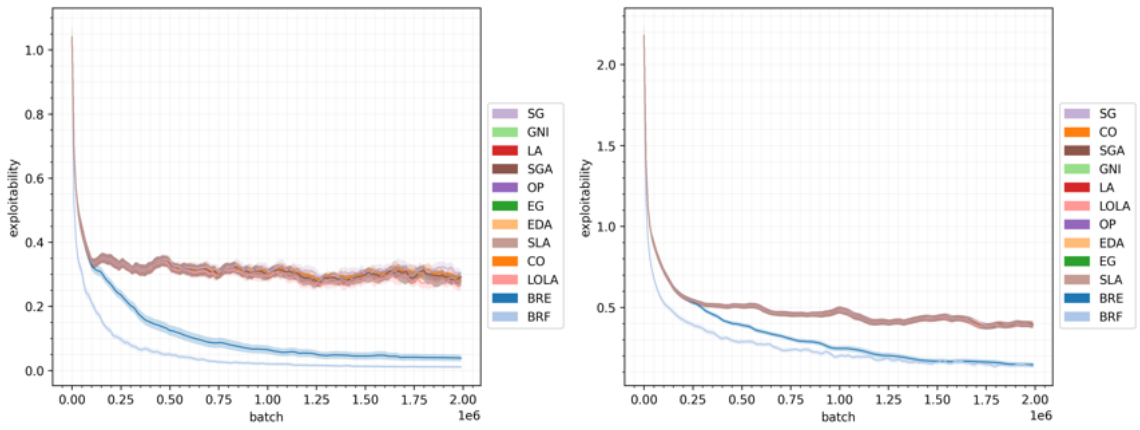


Figure 5.25: Kuhn poker with 2 and 3 players.

5.2.2.8 GAN training

A **generative adversarial network** (GAN) (Goodfellow et al., 2014) is a generative model that consists of two neural networks: a generator and discriminator. The generator maps latent noise to a data sample. The discriminator maps a data sample to a probability. The generator learns to generate fake data, while the discriminator learns to distinguish it from real data. GAN training is a very high-dimensional problem with a highly nonlinear utility function, since the strategies are parameters for the generator and discriminator.

We test the equilibrium-finding methods on the following datasets. The **ring** dataset consists of a mixture of 8 Gaussians with a standard deviation of 0.1 whose means are equally spaced around a circle of radius 1. The **grid** dataset consists of a mixture of 9 Gaussians with a standard deviation of 0.1 whose means are laid out in a regular square grid spanning from -1 to $+1$ in each coordinate. The **spiral** dataset consists of a noisy Archimedean spiral, where $t \sim \mathcal{U}(0,1)$, $r = \sqrt{t}$, $\theta = 2\pi rn$, $x = \mathcal{N}(r\cos\theta, \sigma)$, and $y = \mathcal{N}(r\sin\theta, \sigma)$. Here, n is the number of turns (we use 2) and σ is the standard deviation of the noise (we use 0.05). Finally, the **cube** dataset consists of points sampled uniformly from the edges of a cube and perturbed with Gaussian noise of scale 0.05.

In all cases, the generator's latent noise distribution is a standard Gaussian matching the dimension of the dataset. The generator and discriminator have hidden layers of size 32. We evaluate our method using **Wasserstein distance** (WD) between the real data distribution and the generator's fake data distribution. It is the minimum transportation cost needed to turn one distribution into another. We estimate the WD between the real distribution and generator distribution by taking 1000 samples of each, computing the Euclidean distance matrix, and solving the resulting linear assignment problem.[4]

Our experimental results are shown in Figure 5.26. Our methods outperform the rest. In all cases, SLA diverged to infinity.

We also test on MNIST (Deng, 2012), a dataset of 70,000 28×28 grayscale images of handwritten digits from 0 to 9. The generator and discriminator networks are the same as before, and fully connected, but with the hidden layer size increased to 256 and the noise dimension increased to 32. Due to the larger network size, we use a smaller learning rate of 10^{-4}. Samples are shown in Figure 5.31.

We proposed two new methods that minimize an approximation of the exploitability with respect to the strategy profile. We evaluated these methods in various continuous games, showing that they outperform prior methods. In some cases, our methods converge while all prior methods diverge.

[4]For the linear assignment problem, we use the implementation found in the Python scientific computing library SciPy (Virtanen et al., 2020), which uses a modified Jonker-Volgenant algorithm with no initialization (Crouse, 2016).

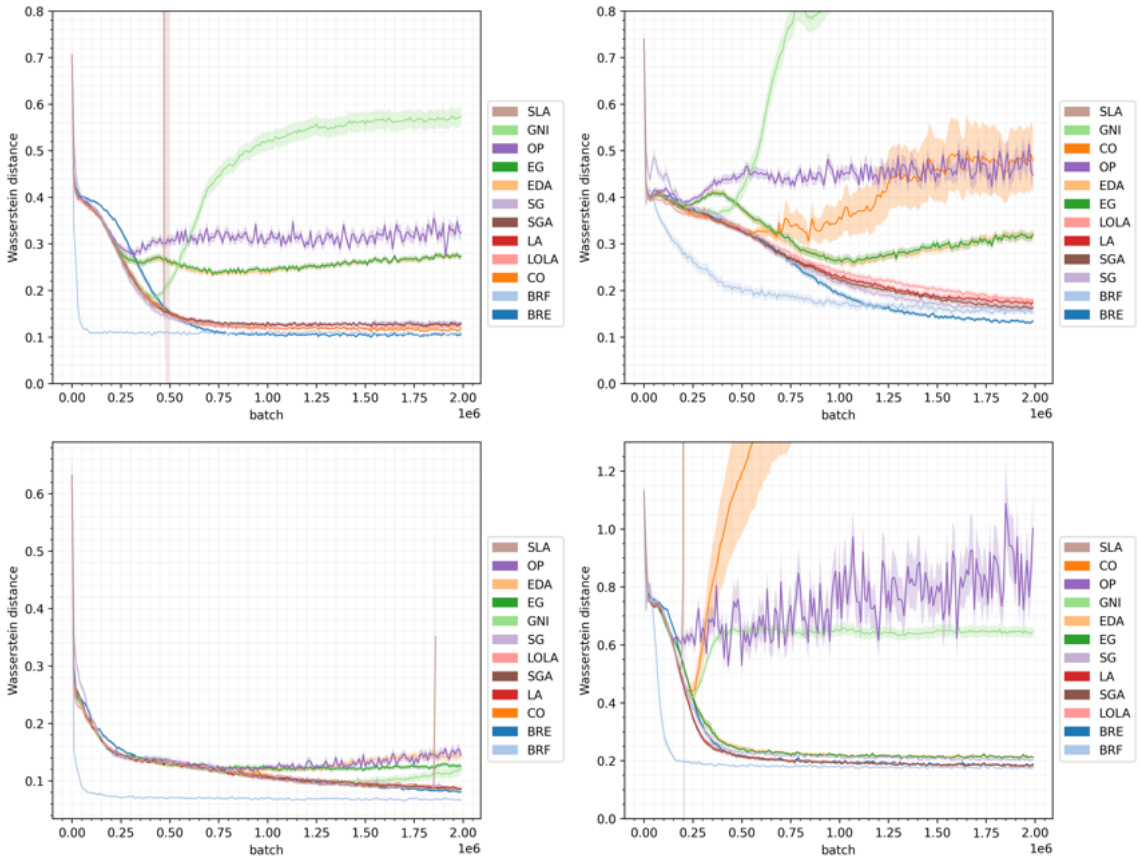


Figure 5.26: Left to right, top to bottom: GAN on ring, grid, spiral, and cube datasets, respectively.

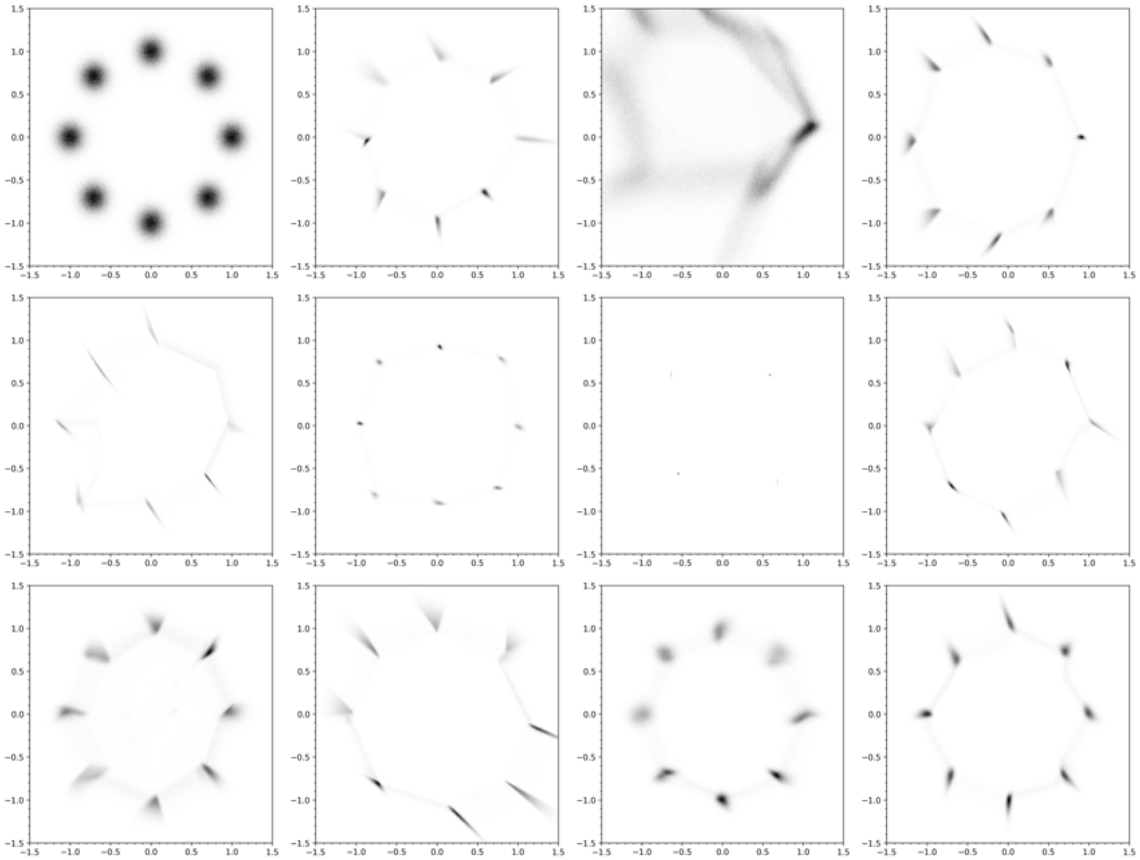


Figure 5.27: GAN (ring dataset). Left to right, top to bottom: Ground truth, SG, OP, EG, CO, SGA, GNI, LA, LOLA, EDA, BRF, BRE. SLA is excluded due to divergence.

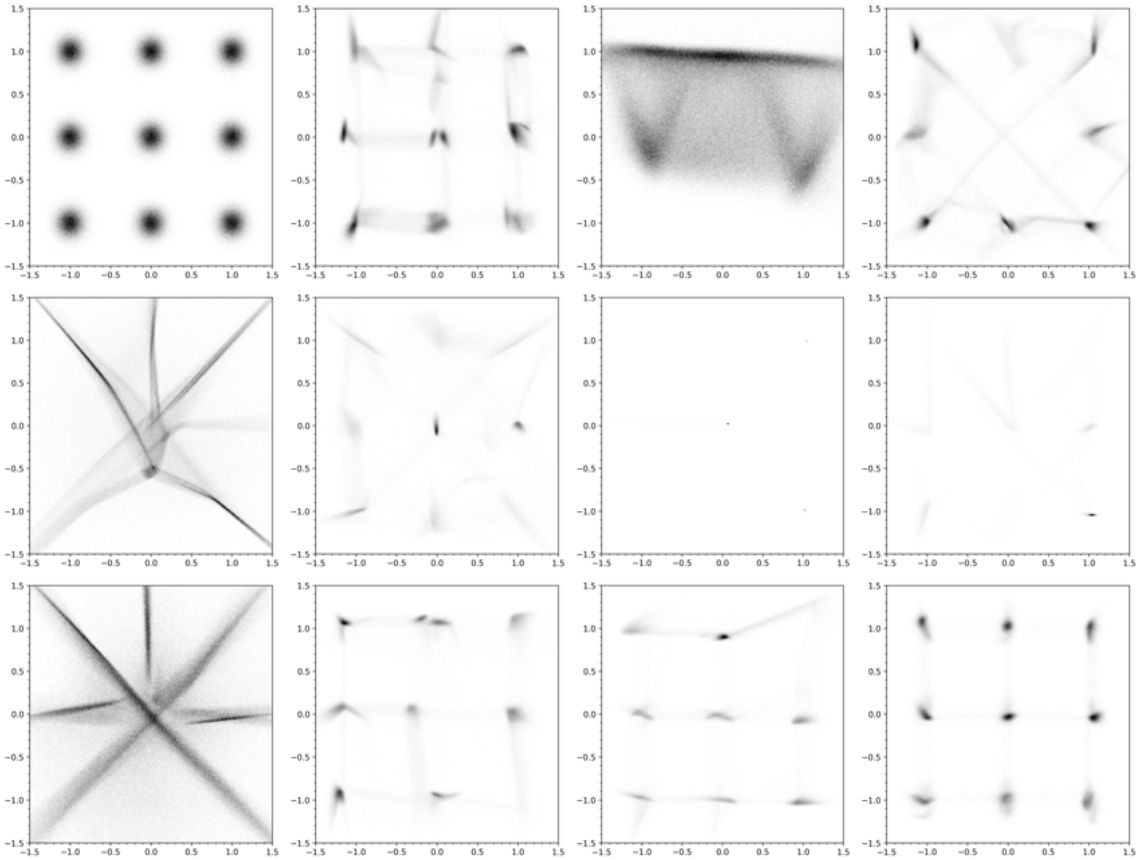


Figure 5.28: GAN (grid dataset). Left to right, top to bottom: Ground truth, SG, OP, EG, CO, SGA, GNI, LA, LOLA, EDA, BRF, BRE. SLA is excluded due to divergence.

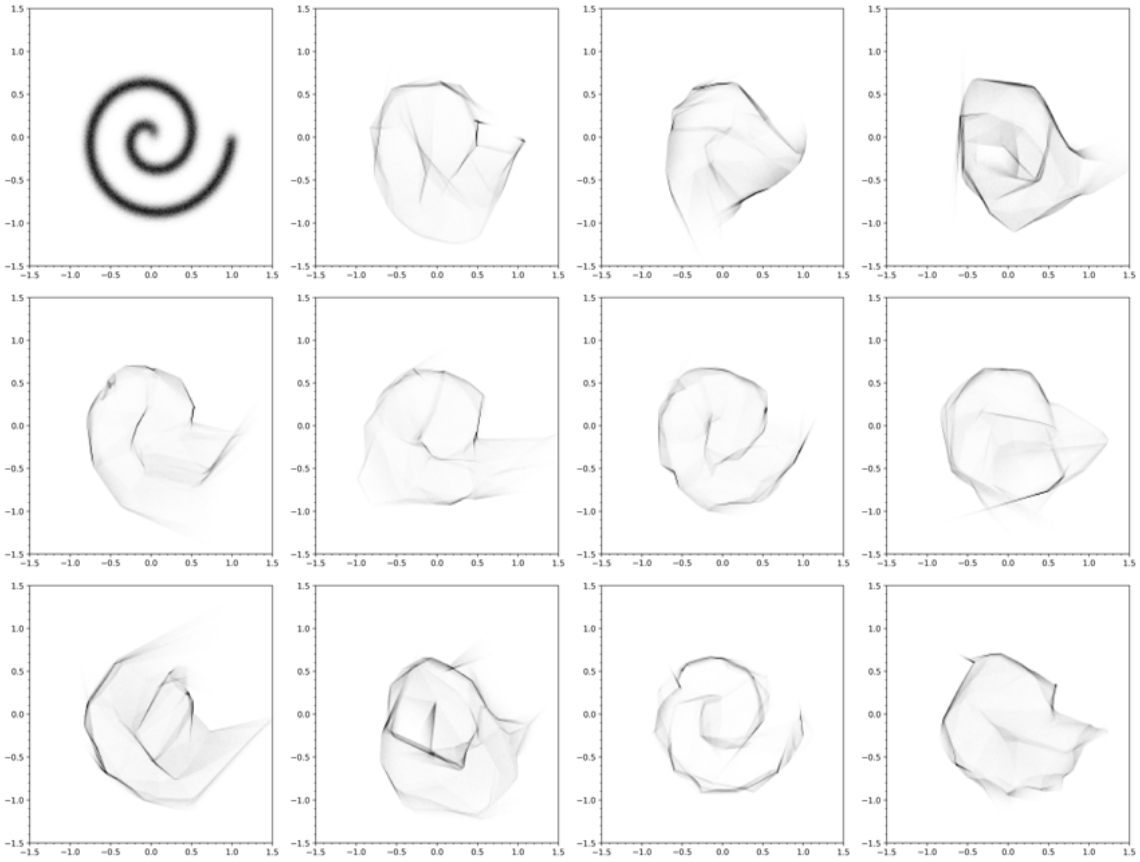


Figure 5.29: GAN (spiral dataset). Left to right, top to bottom: Ground truth, SG, OP, EG, CO, SGA, GNI, LA, LOLA, EDA, BRF, BRE. SLA is excluded due to divergence.

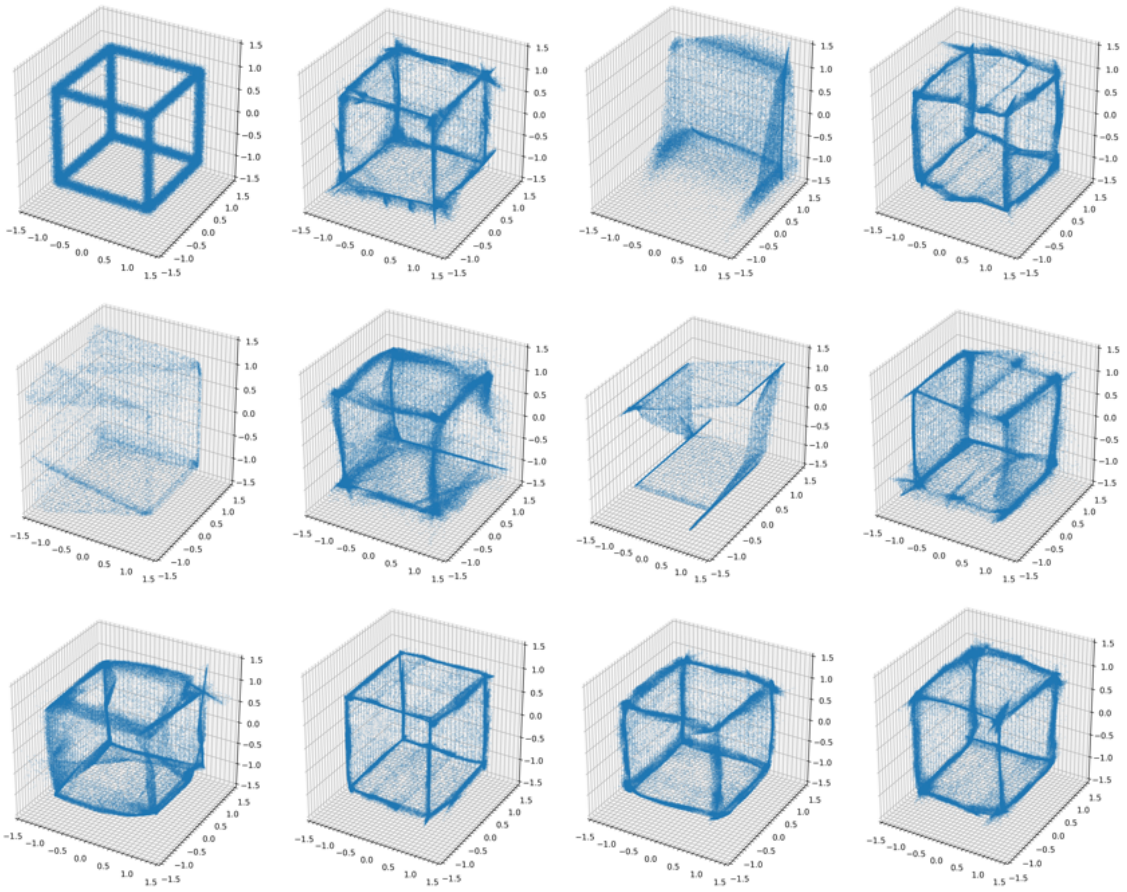


Figure 5.30: GAN (cube dataset). Left to right, top to bottom: Ground truth, SG, OP, EG, CO, SGA, GNI, LA, LOLA, EDA, BRF, BRE. SLA is excluded due to divergence.

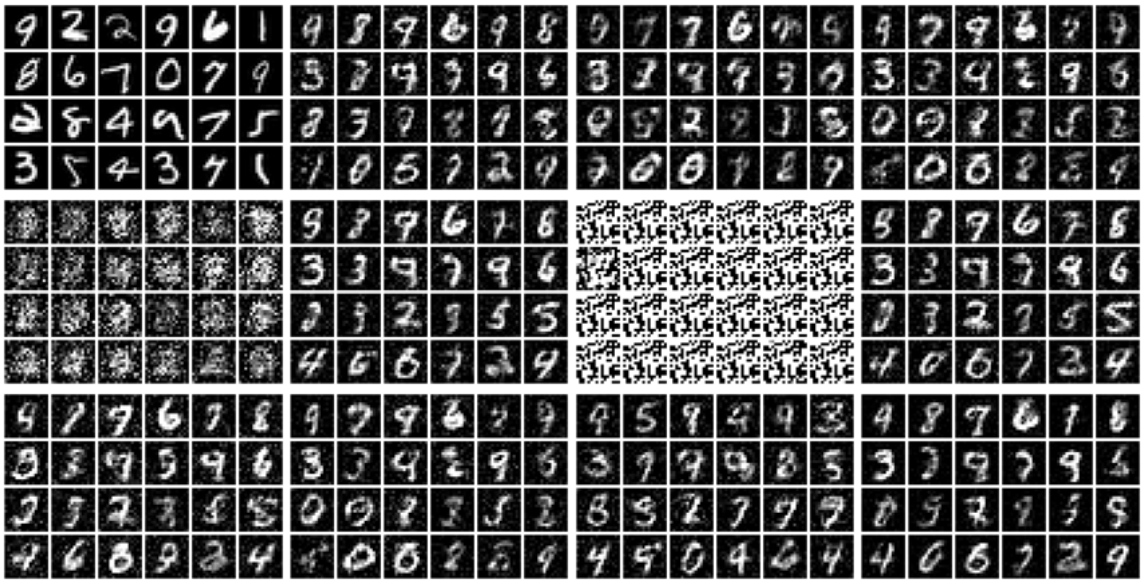


Figure 5.31: GAN (MNIST dataset). Left to right, top to bottom: Ground truth, SG, OP, EG, CO, SGA, GNI, LA, LOLA, EDA, BRF, BRE. SLA is excluded due to divergence.

5.3 Joint-perturbation simultaneous pseudo-gradient

In Martin and Sandholm (2025d), we study the problem of computing an approximate Nash equilibrium of a game whose strategy space is continuous without access to gradients of the utility function. Lack of access to gradients is common in reinforcement learning settings, where the environment is treated as a black box, as well as equilibrium finding in mechanisms such as auctions, where the mechanism's payoffs are discontinuous in the players' actions. To tackle this problem, we turn to zeroth-order optimization techniques that combine pseudo-gradients with equilibrium-finding dynamics. Specifically, we introduce a new technique that requires a number of utility function evaluations per iteration that is constant rather than linear in the number of players. It achieves this by performing a single joint perturbation on all players' strategies, rather than perturbing each one individually. This is very important for many-player games, especially when the utility function is expensive to compute in terms of wall time, memory, money, or other resources. We evaluate our approach on various games, including auctions, which have important real-world applications. Our approach yields a dramatic improvement in performance in terms of the wall time required to reach an approximate Nash equilibrium.

5.3.1 Method

Recall from Section 4.3 that a common approach to game-solving in the literature is **simultaneous gradient ascent** (SG), which is defined as follows. Let $\mathbf{u} \in \mathbb{R}^{n \times d} \to \mathbb{R}^n$ be a utility function, where n is the number of players and d is the dimensionality of each player's strategy parameters. The **simultaneous gradient** of $\mathbf{u}$ is the function $\mathbf{v} \in \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$ where $\mathbf{v}_i = \nabla_i \mathbf{u}_i$. That is, for each player, it is the derivative that player's utility with respect to that player's parameters. Equivalently, $\mathbf{v} = \operatorname{diag} \nabla \mathbf{u}$, where $\nabla \mathbf{u}$ is the Jacobian of $\mathbf{u}$. SG consists of discretizing the ordinary differential equation $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \mathbf{v}(\mathbf{x})$ in time. That is, each player tries to greedily increase their own utility, acting as if the other players are fixed. Explicitly, it uses the iteration scheme $\mathbf{x}_{t+1} = \mathbf{x}_t + \alpha_t \mathbf{v}(\mathbf{x}_t)$ for $t \in \mathbb{N}$, where $\alpha_t > 0$ is some stepsize.

Simultaneous pseudo-gradient. SG and other learning dynamics listed in Section 4.3 require computing the simultaneous gradient $\mathbf{v}$. However, in some situations, $\mathbf{v}$ does not exist because $\mathbf{u}$ is not differentiable. In other situations, $\mathbf{u}$ *is* differentiable, but obtaining an unbiased estimator of its gradient is difficult or intractable. This can happen if, for example, $\mathbf{u}$ is an expectation (with respect to a distribution parameterized by $\mathbf{x}$) of some non-differentiable function. An example of such a situation is an auction with private values. To resolve this problem, we replace the gradient of $\mathbf{u}_i$ in the definition of $\mathbf{v}$ with a pseudo-gradient. Explicitly, $\mathbf{g}_i = \frac{1}{\sigma}\mathbf{u}(\mathbf{x}[i \mapsto \mathbf{x}_i + \sigma \mathbf{z}_i])_i \mathbf{z}_i$ for each Player i, where $\mathbf{z}_i \sim \mu_i$ and μ_i is a multivariate standard normal distribution of the same dimension as $\mathbf{x}_i$. That is, we estimate the pseudo-gradient of $\mathbf{u}_i$ (which is a scalar-valued function, since it outputs only the utility of Player i) with respect to the parameters of Player i. This is the approach taken by Bichler, Fichtl, Heidekrüger, et al. (2021). It requires one perturbation for each player and subsequent evaluation of $\mathbf{u}$. Thus, the number of utility function evaluations per iteration is linear in the number of players.

Pseudo-Jacobian. We extend the preceding concept of the pseudo-gradient from a scalar-valued function to a vector-valued function. Let $n \in \mathbb{N}$, $\mathbf{f} \in \mathbb{R}^d \to \mathbb{R}^n$, and $\mathbf{f}_\sigma(\mathbf{x}) = \mathrm{E}_{\mathbf{z} \sim \mu} \mathbf{f}(\mathbf{x} + \sigma \mathbf{z})$. By

analogy with the pseudo-gradient, we call $\nabla \mathbf{f}_\sigma$ the **pseudo-Jacobian** of $\mathbf{f}$. Furthermore, it satisfies the identity $\nabla \mathbf{f}_\sigma(\mathbf{x}) = \mathrm{E}_{\mathbf{z}\sim\mu} \frac{1}{\sigma}\mathbf{f}(\mathbf{x}+\sigma\mathbf{z}) \otimes \mathbf{z}$. Therefore, we have an unbiased estimator for it.

In the remainder of this subsection, we show that this estimator is not "too noisy". (For example, it is possible in principle for an estimator to be unbiased, but have very large or even infinite variance.) To do this, we give quantitative upper bounds on the moments of the magnitude of this estimator. Suppose we use the central-difference stencil described in Section 2.8. This estimator is $\mathbf{J} = \frac{\mathbf{f}(\mathbf{x}+\sigma\mathbf{z})-\mathbf{f}(\mathbf{x}-\sigma\mathbf{z})}{2\sigma} \otimes \mathbf{z}$ where $\mathbf{z}$ is a sample from the standard d-dimensional multivariate normal distribution. Suppose $\mathbf{f}$ is α-Hölder continuous with constant C. That is, for all $\mathbf{x}, \mathbf{y} \in \operatorname{dom} \mathbf{f}$, $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \le C\|\mathbf{x}-\mathbf{y}\|^\alpha$. Then

$$\|\mathbf{J}\|_{\mathrm{F}} = \frac{\|\mathbf{f}(\mathbf{x}+\sigma\mathbf{z}) - \mathbf{f}(\mathbf{x}-\sigma\mathbf{z})\|}{2\sigma}\|\mathbf{z}\| \tag{5.10}$$

$$\le \frac{C\|2\sigma\mathbf{z}\|^\alpha}{2\sigma}\|\mathbf{z}\| \tag{5.11}$$

$$= (2\sigma)^{\alpha-1}C\|\mathbf{z}\|^{\alpha+1} \tag{5.12}$$

where $\|\cdot\|_{\mathrm{F}}$ is the matrix Frobenius norm. Now, the nth raw moment of the chi distribution is $\mathbb{E}\,\|\mathbf{z}\|^n = 2^{n/2}\Gamma(\frac{d+n}{2})/\Gamma(\frac{d}{2})$, where $d \in \mathbb{N}$ is the dimensionality of $\mathbf{z}$. Thus

$$\mathbb{E}\,\|\mathbf{J}\|_{\mathrm{F}}^n \le (2\sigma)^{n(\alpha-1)}C^n\,\mathbb{E}\,\|\mathbf{z}\|^{n(\alpha+1)} \tag{5.13}$$

$$= (2\sigma)^{n(\alpha-1)}C^n 2^{n(\alpha+1)/2}\Gamma(\tfrac{d+n(\alpha+1)}{2})/\Gamma(\tfrac{d}{2}) \tag{5.14}$$

$$= 2^{n(3\alpha-1)/2}\sigma^{n(\alpha-1)}C^n\Gamma(\tfrac{d+n(\alpha+1)}{2})/\Gamma(\tfrac{d}{2}) \tag{5.15}$$

This yields an upper bound on any moment of the norm of the estimator. It can be applied to C-Lipschitz functions with $\alpha = 1$, and to C-bounded-range functions with $\alpha = 0$.

Joint perturbation. We now combine all of the preceding concepts that have been discussed so far. That is, we combine (1) the identity $\mathbf{v} = \operatorname{diag} \nabla \mathbf{u}$, (2) the concept of the pseudo-Jacobian, and (3) the identity $\operatorname{diag}(\mathbf{a} \otimes \mathbf{b}) = \mathbf{a} \odot \mathbf{b}$ to obtain an estimator that requires only a *single*, joint perturbation across all players. Specifically, let $\mathbf{v}_\sigma = \mathbf{v} * G_\sigma$, that is, the smoothing of $\mathbf{v}$ by a Gaussian kernel of width σ. Then[5]

$$\mathbf{v}_\sigma(\mathbf{x}) = (\mathbf{v} * G_\sigma)(\mathbf{x}) \tag{5.16}$$

$$= ((\operatorname{diag} \nabla \mathbf{u}) * G_\sigma)(\mathbf{x}) \tag{5.17}$$

$$= \operatorname{diag}(\nabla \mathbf{u} * G_\sigma)(\mathbf{x}) \tag{5.18}$$

$$= \operatorname{diag} \nabla(\mathbf{u} * G_\sigma)(\mathbf{x}) \tag{5.19}$$

$$= \operatorname{diag} \nabla \mathbf{u}_\sigma(\mathbf{x}) \tag{5.20}$$

$$= \operatorname{diag} \mathrm{E}_{\mathbf{z}\sim\mu} \tfrac{1}{\sigma}\mathbf{u}(\mathbf{x}+\sigma\mathbf{z}) \otimes \mathbf{z} \tag{5.21}$$

$$= \mathrm{E}_{\mathbf{z}\sim\mu} \tfrac{1}{\sigma} \operatorname{diag} \mathbf{u}(\mathbf{x}+\sigma\mathbf{z}) \otimes \mathbf{z} \tag{5.22}$$

$$= \mathrm{E}_{\mathbf{z}\sim\mu} \tfrac{1}{\sigma}\mathbf{u}(\mathbf{x}+\sigma\mathbf{z}) \odot \mathbf{z} \tag{5.23}$$

Therefore, we obtain the following unbiased estimator: $\mathbf{g} = \frac{1}{\sigma}\mathbf{u}(\mathbf{x}+\sigma\mathbf{z}) \odot \mathbf{z}$. In terms of indices, for clarity: $\mathbf{g}_i = \frac{1}{\sigma}\mathbf{u}(\mathbf{x}+\sigma\mathbf{z})_i\mathbf{z}_i$. With this new estimator, the number of utility function evaluations per

[5]The interchange of differentiation and integration is justified by the Hölder continuity of $\mathbf{u}$ combined with Gaussian smoothing (Duchi, Bartlett, and Wainwright, 2012).

iteration is now *constant* in the number of players, rather than linear. This dramatically reduces the number of utility function evaluations when there are many players. Therefore, it makes game solving significantly more efficient in many-player games, especially when the utility function is expensive to evaluate in terms of wall time, memory, money, or other resources such as real-world experiments. We call our method **joint-perturbation simultaneous pseudo-gradient** (JPSPG), in contrast to the classical method, **simultaneous pseudo-gradient** (SPG). To our knowledge, this is the first work to define the concept of the pseudo-Jacobian and apply it to the estimation of the simultaneous gradient. The intuition behind our method is that we estimate the analogue of the gradient (i.e., Jacobian) of the *vector*-valued function that returns utilities for all n players *simultaneously* (and then select its diagonal), rather than treating the problem as having n different, separate *scalar*-valued functions whose gradients need to be estimated separately. This saves us work, because we need just 1 rather than n different function evaluations. For clarity, we emphasize that our method does not assume or require symmetry across players. It handles general utility functions on general strategy spaces. The utility function does not need to be symmetric across players. In fact, the players' strategy spaces need not be equal. For example, player 1's parameters could be 10-dimensional, player 2's parameters could be 20-dimensional, and so on. One can use our method in combination with any gradient-based learning dynamics from the literature that are based on the simultaneous gradient. These include learning dynamics such as standard simultaneous gradient ascent, extragradient ascent (Korpelevich, 1976), optimistic gradient ascent (Popov, 1980; Daskalakis et al., 2018), and so on. In the games that we tested on, simultaneous gradient ascent was sufficient to obtain convergence to an approximate NE. However, we can combine JPSPG with any other learning dynamics.

5.3.2 Experiments

We test our approach against the classical approach on several continuous-action games, in particular on many kinds of auction and on continuous Goofspiel (which can be thought of as a kind of sequential auction with budget constraints). These are many-player games where each utility function evaluation is expensive, thus highlighting the problem of interest and showing the benefits of our method. Specifically, each utility function evaluation requires solving a linear assignment problem, solving an integer linear program, running policies over multiple steps, etc., all of which are expensive operations. The games we test on cover various theoretical properties of interest, including various dimensionalities for each player's observation, various dimensionalities for each player's action, single-step vs. multi-step settings, etc. Our experimental hyperparameters are as follows. For each experiment, we run 8 trials. In each graph, solid lines show the mean across trials, and bands show the standard error of the mean. The classical method is shown in blue, while our method is shown in orange. We use a stepsize of 10^{-4}. For the Gaussian smoothing, we use a perturbation scale σ of 10^{-1}. We use a batch size per iteration of 256. To update parameters, we use the AdaBelief optimizer (Zhuang et al., 2020). For each player's strategy network, we use a single hidden layer of size 64, the ReLU activation function, and He initialization (He et al., 2015) for initializing the network's weights.

5.3.2.1 Multi-item unit-demand auction

Here, we consider a type of multi-item auction called a **unit-demand auction**. In this auction, we have n bidders and m non-identical items. Each bidder i has a private valuation v_{ij} for each item j. Furthermore, each bidder has *unit demand*, meaning that its value for a *bundle* of items

is the same as that for the maximum-value item in that bundle: $v_i(S) = \max_{j \in S} v_{ij}$, where S is a bundle of items. Housing markets are often given as an example of unit-demand preferences. This model was first studied by Shapley and Shubik (1971). This is a special case of a **limited-demand model** with K units in which each bidder has use for at most $L < K$ units, as described in Krishna (2002, §13.4.2 and §13.5.2). The single-unit case corresponds to $L = 1$. For our experiment, we use a prior where bidder-item valuations v_{ij} are independently sampled uniformly at random from the unit interval. Each player i submits a bid b_{ij} for each item j. To allocate items, our auction mechanism assigns items to bidders in a way that maximizes the sum of bids across players. This requires solving a **linear assignment problem**, which can be described as follows. Given a bid matrix $b \in \mathbb{R}^{n \times m}$, compute a binary assignment $x \in \{0, 1\}^{n \times m}$ that satisfies the following:

$$\text{maximize} \quad \sum_{i \in [n]} \sum_{j \in [m]} b_{ij} x_{ij} \tag{5.24}$$

$$\text{subject to} \quad \sum_{i \in [n]} x_{ij} \leq 1 \qquad \forall j \in [m] \tag{5.25}$$

$$\sum_{j \in [m]} x_{ij} \leq 1 \qquad \forall i \in [n] \tag{5.26}$$

That is, maximize the sum of values subject to the constraint that each item is assigned to at most one bidder and each bidder receives at most one item. An example is illustrated in Figure 5.32.

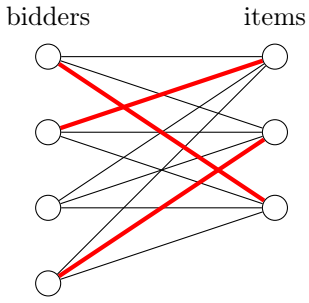


Figure 5.32: Example of a complete bipartite graph between bidders and items, and an assignment.

The linear assignment problem was first described in a seminal paper by Kuhn (1955), who introduced a solution approach called the **Hungarian method**. Subsequently, various algorithms have been devised in the literature. We use the modified **Jonker-Volgenant algorithm** (Jonker and Volgenant, 1988) given in Crouse (2016), as implemented in the Python scientific computing library SciPy (Virtanen et al., 2020), and reimplement it in JAX (Bradbury et al., 2018). This algorithm has a time complexity of $O(n^3)$.

Figure 5.33 shows the exploitability over the course of training for a unit-demand auction. Both our method and the classical method attain a similar exploitability for a given iteration count, but our method is dramatically faster in terms of run time (here expressed in seconds). This is explained by the fact that the standard method requires many more evaluations of the utility function per iteration, each of which requires solving an assignment problem (which is expensive), whereas ours only requires a *single* utility function evaluation. The advantage of our method over the baseline increases as the number of players increases.

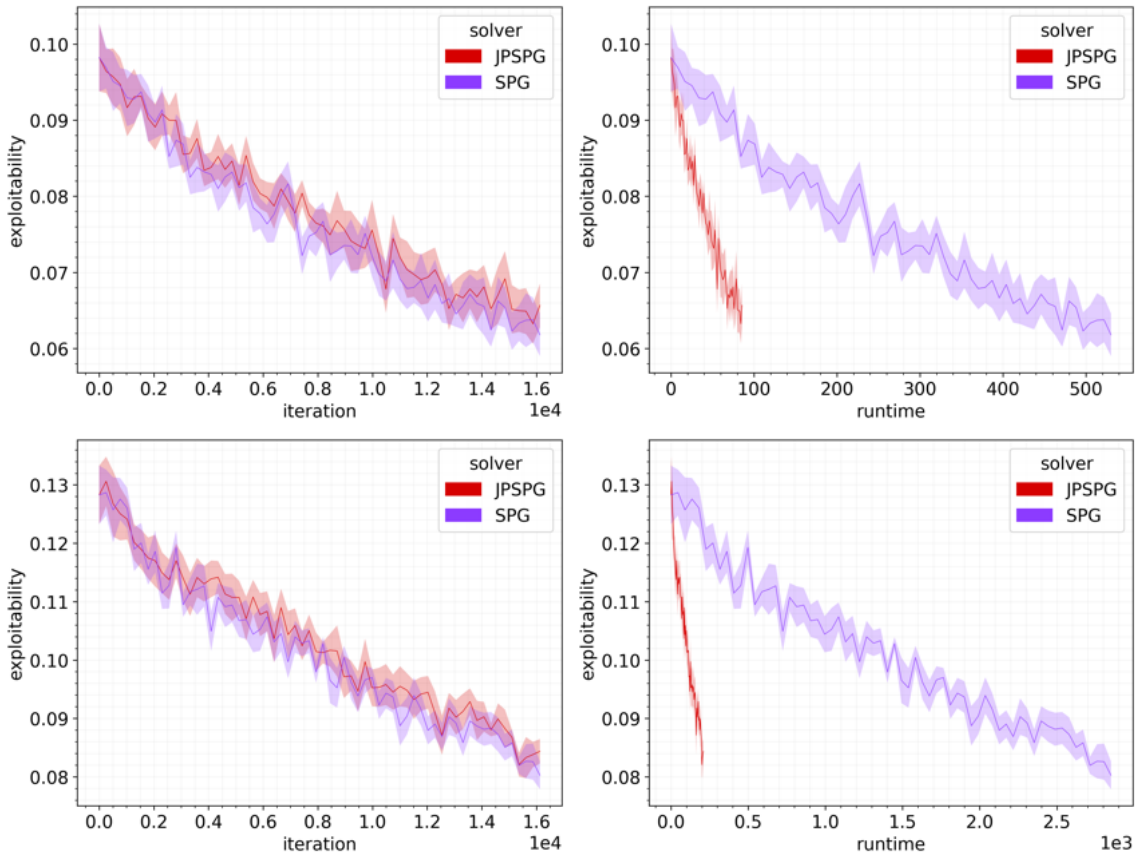


Figure 5.33: Unit-demand auction. Top: 10 players, 10 items. Bottom: 20 players, 20 items.

5.3.2.2 Knapsack auction

Another type of auction is a **knapsack auction**. We follow the description given in Aggarwal and Hartline (2006). In a knapsack auction, an auctioneer auctions off space in a knapsack of known capacity C. Each player seeks to place exactly one object in the knapsack. Player i values the placement of its object in the knapsack at v_i. The valuations are private data of each respective player. Each object takes up a certain amount of space in the knapsack. Player i's object takes space c_i, and these sizes are publicly known. Thus, the c_is are public while the v_is are private. Each player submits a bid b_i. Among other works, knapsack auctions have been studied by Dütting, Gkatzelis, and Roughgarden (2014) and Berg et al. (2010), who model the problem of bidding in ad auctions as a penalized multiple choice knapsack problem. As Aggarwal and Hartline (2006) note, the knapsack auction problem models several interesting applications. For example, consider running a single auction to sell advertising space on a web page over the course of a day. Suppose statistical information is available for each advertiser as to how many showings (i.e., impressions) are necessary to result in a user click-through and as well as how many times the web page itself will

be viewed in a day. The number of impressions necessary to generate a click-through corresponds to the c_is and the number of total views corresponds to the capacity of the knapsack, C. Each utility function evaluation requires solving an optimization problem of the following form: maximize $\mathbf{x} \cdot \mathbf{b}$ subject to $\mathbf{x} \cdot \mathbf{c} \leq C$ and $\mathbf{x} \in \{0, 1\}^n$. Here, n is the number of players, $\mathbf{b}$ is the vector of stated values (bids) for each player, $\mathbf{c}$ is the corresponding vector of sizes, $\mathbf{x}$ is a binary vector indicating whether each player is included in the knapsack. Player i's final utility is $(v_i - b_i)x_i$. That is, it is the difference between their private value and their submitted bid, assuming they are included in the knapsack, and zero otherwise. This problem can be solved using **integer linear programming** (ILP). For this, we use the `milp` function included in SciPy's library (Virtanen et al., 2020), which uses the HiGHS optimization solver (Huangfu and Hall, 2018; Hall et al., 2023). Solving an integer program can be expensive (integer linear programming is NP-hard), so reducing the number of utility function evaluations during learning should result in a significant speedup. In our experiment, we sample the v_is and c_is from the standard uniform distribution, and sample C from the standard uniform distribution on $[0, n]$. Experimental results on the knapsack auction are shown in Figure 5.34. Our method requires fewer utility function evaluations per iteration and thus yields a dramatic improvement in training time for attaining the same exploitability.

5.3.2.3 Sequential auction for identical items

Consider a multi-item unit-demand auction in which identical items are sold *sequentially*, rather than simultaneously. We follow the description given in Krishna (2002, §15.1). In this auction, K identical items are sold to $N > K$ bidders using a sequence of first-price sealed-bid auctions. Specifically, on each of K rounds, one of the items is auctioned using the first-price format, and the price at which it is sold—the winning bid—is announced. We focus on the **single-unit demand** setting, in which each bidder has use for at most one unit. Thus a bidder leaves the game once it has won an item. Each bidder has a private value v_i that is sampled from the standard uniform distribution. On round k, a bidder's observation consists of its own private value as well as all the prices of the preceding $k - 1$ rounds, $p_1, p_2, \ldots, p_{k-1}$. Results are shown in Figure 5.35. Our method yields a dramatic improvement in terms of the wall time required to reach a certain level of exploitability.

5.3.2.4 Continuous-action Goofspiel

Goofspiel, also known as *the game of pure strategy*, is a card game invented by mathematician Merrill Flood in the 1930s (Tucker, 1984). This game is played with a standard card deck. The cards of one suit are given to one player, the cards of a second suit are given to the other player, and the cards of a third suit are shuffled and placed face down in the middle. The cards are valued from low to high as 1 (Ace), 2, 3, ..., 10, 11 (Jack), 12 (Queen), and 13 (King). A round consists of turning up the next card from the middle pile and letting the players simultaneously "bid" on this "prize" card. Players bid by choosing one of their own cards and revealing it at the same time as the other player. The player with the highest bid wins the value of the prize card. In a tie, the prize value is split between the players. All three cards are then discarded. The game ends after 13 rounds, and the winner is the player with the highest score. Because of its simple mechanics but complex strategy, Goofspiel is commonly used as an example in game theory and artificial intelligence, and has been studied extensively in the literature (Ross, 1971; Dror, 1989; Ferguson and Melolidakis, 2001; Grimes and Dror, 2013; Rhoads and Bartholdi, 2012; Lanctot, Lisý, and Winands, 2014). We consider the following *continuous-action* variant of Goofspiel. Instead of receiving a deck of discrete bids consisting of all cards of one suit, each player has a *continuous* budget that they can spend to

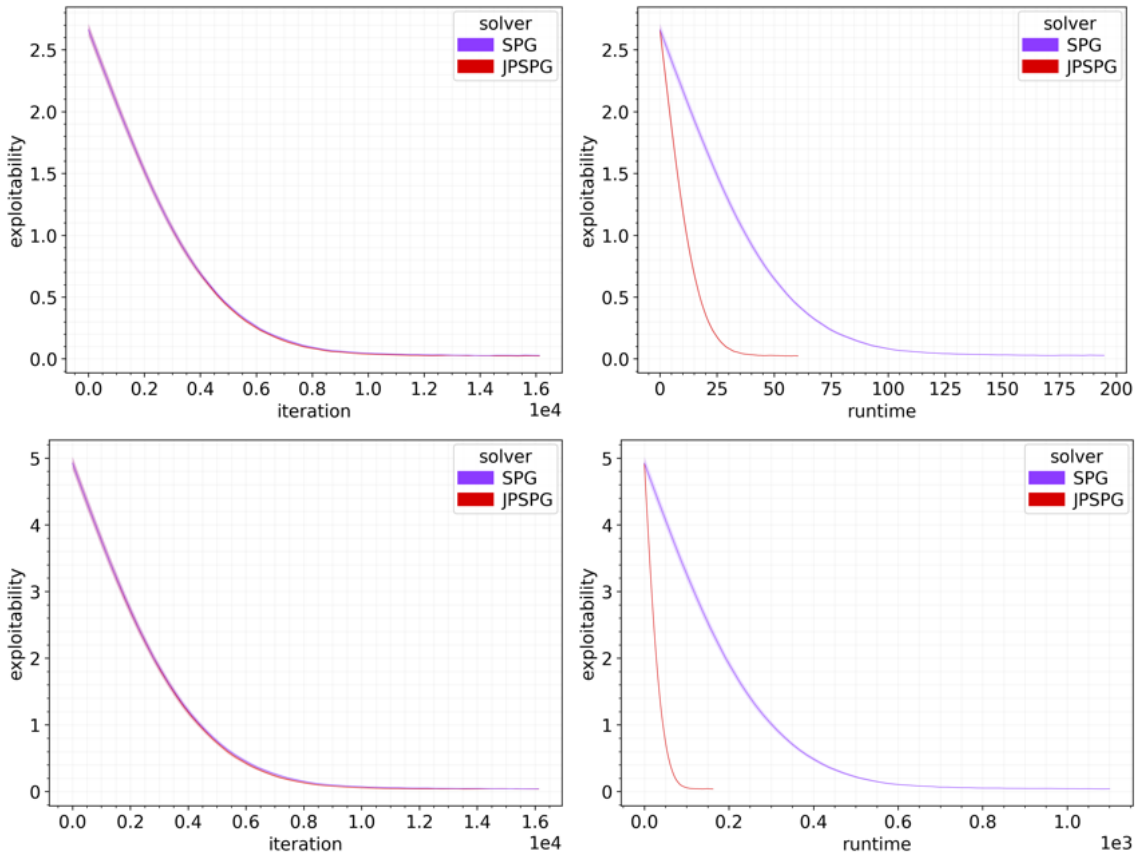


Figure 5.34: Knapsack auction. Top: 10 players. Bottom: 20 players.

bid on the prize card in each round. In each round, their continuous bid is subtracted from their budget. This can be thought of as a multi-round, multi-item, auction-like scenario with a budget constraint for each bidder. To allow the players to randomize over their 1-dimensional actions (the bids), we inject their strategy networks with 1-dimensional latent input noise in addition to the observation, as described in Martin and Sandholm (2023). Figure 5.36 shows the exploitability over the course of training on continuous-action Goofspiel. Our method yields a significant improvement in the run time required to attain a certain level of exploitability.

Our experimental results confirm our hypothesis, namely, that our approach yields a dramatic improvement in the training time required to reach a certain level of exploitability.

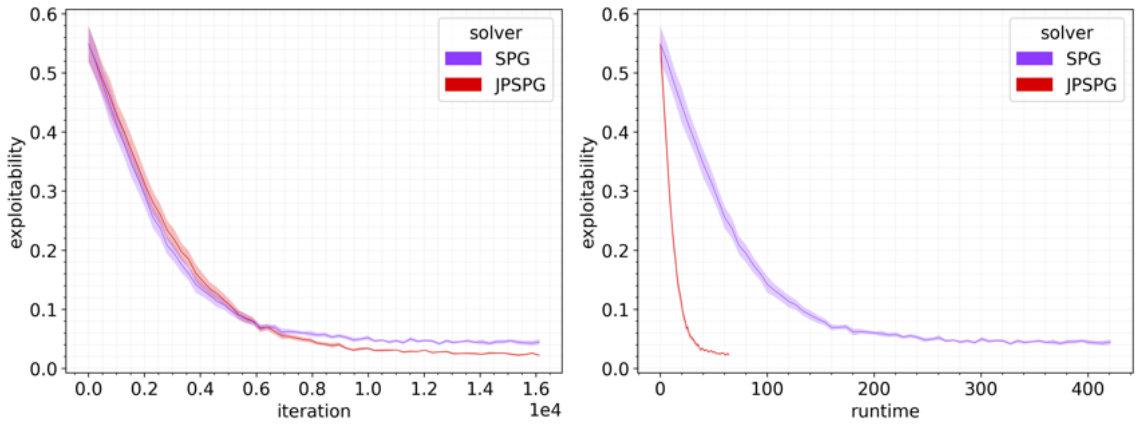


Figure 5.35: 20-player, 10-item sequential auction.

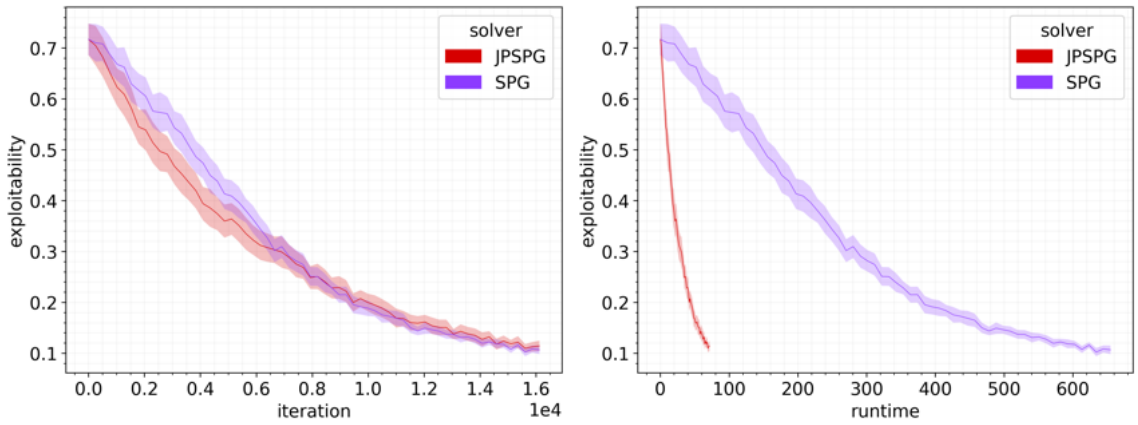


Figure 5.36: 20-player continuous Goofspiel.

5.4 Solving infinite-player games with player-to-strategy networks

In Martin and Sandholm (2025e), we present a new approach to solving games with a countably or uncountably infinite number of players. Such games are often used to model multiagent systems with a large number of agents. The latter are frequently encountered in economics, financial markets, crowd dynamics, congestion analysis, epidemiology, and population ecology, among other fields. Our two primary contributions are as follows. First, we present a way to represent strategy profiles for an infinite number of players, which we name a Player-to-Strategy Network (P2SN). Such a network maps players to strategies, and exploits the generalization capabilities of neural networks to learn

across an infinite number of inputs (players) simultaneously. Second, we present an algorithm for training such a network to find approximate Nash equilibria, which we name Shared-Parameter Simultaneous Gradient (SPSG). This algorithm generalizes simultaneous gradient ascent and its variants, which are traditionally used for multiagent reinforcement learning. We test our approach on infinite-player games and observe its convergence to approximate Nash equilibria. Our method can handle games with infinitely many players, infinitely many actions (and mixed strategies over them), infinitely many states, and discontinuous player-specific utility functions. To our knowledge, this is the first paper to tackle general infinite-player games using neural networks.

5.4.1 Method

Our method consists mainly of two parts: Strategy profile representation and training.

P2SN. Suppose we are interested in tackling a game with infinitely many players. This raises the problem of how to represent a strategy profile that has an infinite number of players. We propose a way to do this, which we call a **Player-to-Strategy Network** (P2SN). This is a neural network that takes as input a *player* and outputs a *strategy* for that player. We exploit the strong generalization capabilities of neural networks (as described in Section 1.1) to represent (and potentially learn, as we will see shortly) across an infinite number of possible inputs, i.e., players, simultaneously.[6] What exactly the network receives as input depends on the game. The inputs include (1) the features identifying the player, (2) any observations the player receives, and (3) random noise that allows the network to randomize over actions (Martin and Sandholm, 2023). This is illustrated in Figure 5.37.

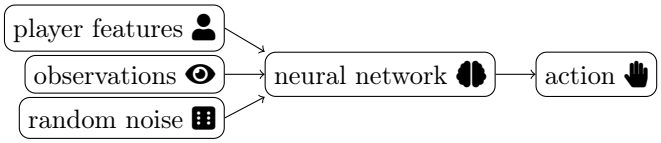


Figure 5.37: Structure of a P2SN.

Player representation. A specific player is input into the P2SN as a set of *player features*. The exact nature of these features depends on the game. For example, in a spatial game, each player might be identified with a specific point in two-dimensional space. Thus a player can be represented as an ordered pair of X and Y coordinates, i.e., a 2D vector. In a game with a continuum of indistinguishable players, such as a continuum of traders in a stock market (Aumann, 1964), the players can be represented simply as points on the unit interval (i.e., scalars), with no additional distinguishing features. In other games, if there is some prior notion of *similarity* between players, they can be represented as points in some embedding space, such that similar players get nearby embeddings and dissimilar players get distant embeddings. This type of learning, which pulls together similar objects and pushes apart dissimilar objects, is known as contrastive learning (Jaiswal et al., 2020; Le-Khac, Healy, and Smeaton, 2020; Hu, Wang, et al., 2024).

[6] P2SN could be useful even in games with a finite number of players. For instance, the input could be a one-hot encoding of the player's index, along with any other *a priori* features of that player. It allows for neural net parameters to be shared across players, while simultaneously taking into account each player's individual and specific *a priori* features as input. This can potentially result in faster learning and better generalization. However, the present paper focuses on games with an infinite number of players.

Fourier features. The space of player feature vectors is often low-dimensional. For example, it is 2-dimensional in the case of a spatial game on the square. It is important for our P2SN to be able to represent fine details on such a space. It has been found that standard feedforward neural networks struggle to learn detailed, high-frequency features in low-dimensional input spaces. Rahaman et al. (2019) showed that standard neural networks are biased towards low-frequency functions, meaning that they cannot have local fluctuations without affecting their global behavior. More precisely, using Fourier analysis, they showed that these networks prioritize learning the low-frequency modes, a phenomenon they call *spectral bias*. Tancik et al. (2020) showed that passing input points through a simple **Fourier feature mapping** enables a **multilayer perceptron** (MLP) to learn high-frequency functions in low-dimensional problem domains, whereas a standard MLP has impractically slow convergence to high-frequency signal components. To allow the P2SN to more easily represent and learn fine details in low-dimensional spaces, which we found to help performance, we pre-process the inputs $\mathbf{x} \in \mathbb{R}^d$ with the map $f(\mathbf{x}) = (\sin(\mathbf{a}), \cos(\mathbf{a}))$, where $\mathbf{a} = \mathbf{B}\mathbf{x} + \mathbf{b}$, $n \in \mathbb{N}$ is the number of Fourier features, $\mathbf{B} \in \mathbb{R}^{d \times n}$ is a learned frequency matrix, and $\mathbf{b} \in \mathbb{R}^n$ is a learned phase vector. Thus f maps the inputs to a high-dimensional space of Fourier features. We initialize $\mathbf{B}$ with independent samples from the normal distribution with standard deviation $\sigma = 100$. This large σ is recommended by Tancik et al. (2020) and yields high spatial frequencies from the start, which facilitates representation and learning. We initialize the entries of $\mathbf{b}$ from the uniform distribution on $[0, 2\pi)$.

Learning dynamics for shared parameters. If parameters are shared between players (instead of maintaining individual disjoint parameters for each player), the original expression for $v(s)_i$ no longer makes sense. This is because it requires taking a derivative with respect to the parameters *of a specific player and no other*, namely s_i, which is no longer possible.

Naive SG. To get around this, we could, for each player, simply take derivatives with respect to *all* parameters: $\tilde{v}(s)_i = \frac{\mathrm{d}u(s)_i}{\mathrm{d}s}$. However, this does not work. For example, consider a two-player zero-sum game where both players share parameters $s \in [0, 1]^2$, player 1's strategy is s_1, player 2's strategy is s_2, and the utility for player 1 is $s_1 + s_2$. If we just ignore the disjointness requirement, the parameters do not change at all, because they are "pulled" in exactly equal and opposite directions. We demonstrate this in Section 5.4.2.1.

Shared-Parameter Simultaneous Gradient (SPSG). To solve these problems, we re-express the simultaneous gradient as a derivative with respect to *the entire strategy profile* as a unit: $v(s) = \left[\frac{\mathrm{d}}{\mathrm{d}r} \int_{i \sim \mu} u(s[i \mapsto r_i])_i\right]_{r=s}$. Here, $[\cdot]_{r=s}$ means evaluating the expression inside the brackets, treated as a function of r, at the argument value s. The derivative of the integral is a **functional derivative**, i.e., the derivative of a functional (the integral) with respect to a function (in this case, r). In contrast to the original expression for the simultaneous gradient, this expression *can* be generalized to the shared-parameter case. Let $s \in \Theta \to \prod_{i \in P} S_i$ be a parameterization of a strategy profile—such as a P2SN—with parameters in Θ. Define $v(\theta) = \left[\frac{\mathrm{d}}{\mathrm{d}\phi} \int_{i \sim \mu} u(s_\theta[i \mapsto (s_\phi)_i])_i\right]_{\phi=\theta}$. We call this the **Shared-Parameter Simultaneous Gradient** (SPSG).

Implementation. In practice, SPSG can be estimated as follows. On each iteration, do the following. First, create a copy ϕ of the current parameters θ. Second, sample a player i from μ.

Algorithm 2 Training a P2SN with SPSG.

Input: Initial parameters $\theta \in \mathbb{R}^d$, learning rate $\eta \in \mathbb{R}$.
$f(\theta, \phi)_i = u(s_\theta[i \mapsto (s_\phi)_i])_i$
for $t \leftarrow 0, 1, 2, \ldots$ **do**
  $i \sim \mu$ ▷ sample a player from the player distribution
  $g \leftarrow \nabla_2 f(\theta, \theta)_i$ ▷ get unbiased estimator of the SPSG
  $\theta \leftarrow \theta + \eta g$ ▷ update the shared parameters

Third, create a hybrid strategy profile $s_\theta[i \mapsto (s_\phi)_i]$ that intertwines s_θ and s_ϕ, using s_ϕ's output to define player i's strategy and s_θ's output to define any other player's strategy. Fourth, pass this strategy profile to the utility function $u(\cdot)_i$ and get player i's utility. Fifth, take the gradient of this scalar with respect to ϕ.[7] The result is an unbiased estimator of $v(\theta)$. The sampling of players yields an unbiased estimator of the entire integral, which in turn yields an unbiased estimator of the SPSG, since the integral and derivative commute.[8] All we need for this procedure is access to $u(s)_i$, or an unbiased estimator thereof. This shown in Algorithm 2.

Estimating integrals. In cases where $u(s)_i$ is an integral over players of some other function, which is sometimes the case in infinite-player games, we can use an unbiased estimator for it. Such an estimator often exists for infinite-player games in practice. For example, many infinite-player games can be expressed as a series of the following general form.

$$u(s)_i = f_i(s_i) + \int_{j \sim \nu_i} \left(g_{ij}(s_i, s_j) + \int_{k \sim \xi_{ij}} h_{ijk}(s_i, s_j, s_k) + \ldots \right) \tag{5.27}$$

where f, g, etc. are arbitrary functions and ν_i, ξ_{ij}, etc. are arbitrary measures. That is, player i's utility is an integral over pairwise interactions with other players, 3-way interactions with other pairs of players, etc., up to some order. In that case, an unbiased estimator of player i's utility can be obtained by first sampling j given i, then sampling k given i and j, and so on, for however many terms are necessary. If necessary, we can use techniques like **Markov chain Monte Carlo** (MCMC) (Metropolis et al., 1953; Hastings, 1970) to obtain these samples. As we will see in the experiments section, many games of interest require only the first integral, i.e., involve only the aggregation of pairwise interactions between players (and their corresponding strategies). We emphasize that our method makes no assumption of symmetry or identicality across players. Each player can have its own distinct strategy space and utility function. For example, in spatial games (where players occupy different points in space), the players are clearly not identical: some are close to the boundary (where boundary effects come into play), and others are not. Furthermore, the computational complexity of our method, when applied to the case where P is finite, is close to that of SGA, differing only in that a *single* player is *stochastically* sampled at each iteration (and used to estimate the SPSG). P2SN and SPSG leverage the generalization capabilities of neural networks to learn across an infinite number of players simultaneously.

[7] Not θ. Though θ and ϕ have equal *values*, they are different *variables*. Consider $f(x, y) = xy^2$. At $x = y = 1$, x and y have the same value. However, the gradient with respect to x is 1, while the gradient with respect to y is 2.

[8] Here, we assume that the conditions of the Leibniz integral rule, which allow differentiation and integration to be exchanged, hold, as described in Talvila (2001).

5.4.2 Experiments

We evaluated the proposed techniques using computational experiments. For each experiment, we ran 16 trials. In each plot of exploitability, solid lines show the mean across trials, and bands show a confidence level of 0.95 for the mean, which is computed using bootstrapping (Efron, 1979), specifically the **bias-corrected and accelerated** (BCa) method (Efron, 1987). The legend entries are sorted by final values. In the neural net, on each iteration, we use a batch size of 64, averaging gradients across that batch. For the P2SN, we use a multilayer perceptron with various depths (number of hidden layers) and widths (number of Fourier features, as well as size of each hidden layer). If not stated explicitly, we use a default depth of 1 and width of 64. We use the ReLU activation function and He weight initialization (He et al., 2015). We use a vanilla SGD optimizer with learning rate as stated in the legends (labeled "lr").

Exploitability estimation. To evaluate performance, we estimate the exploitability. To do this, we discretize the space of players into N_{players} points[9], estimate the exploitability of each individual player, and average. To estimate an individual player's exploitability, we estimate the quantities described in the formulation of Bayesian games at the end of Section 3.3. Specifically, we sample $N_{\text{observations}}$ observations from that player's observation distribution to average over, discretize the action space into N_{actions} points to maximize over, and sample N_{samples} utilities conditioned on the observation to average over.[10] We use $N = 256$ for each of these.

For brevity, we include only some experiments here. The rest can be found in Martin and Sandholm (2025e).

5.4.2.1 Sum game

To illustrate why Naive SG fails, and why we need SPSG instead, we use a very simple game. This is a two-player zero-sum game in which both players simultaneously choose values on the unit interval, and the utility for the first player is the sum of the values. Formally, $P = \{1, 2\}$, $S(p) = [0, 1]$, $u(s, 1) = -u(s, 2) = s(1) + s(2)$. We run training with the two different dynamics. Exploitabilities are shown in Figure 5.38. As expected, the naive dynamics fail to converge, because the parameters are pulled in equal and opposite directions (with some noise from which player is sampled in each iteration). On the other hand, SPSG does converge.

5.4.2.2 Anti-coordination game

Let $d \in \mathbb{N}$ be a spatial dimension. Let $r \in \mathbb{R}$ be an interaction radius. Let $k \in \mathbb{N}$ be a number of resources. Let $b \in \mathbb{R}^d \to \mathbb{R}^k$ be a bias field. The player set is $P = [0, 1]^d$. The strategy set function is $S(x) = \triangle[k]$. The utility function is $u(s, x) = s(x) \cdot (b(x) - \bar{s}(x))$, where $\bar{s}(x) = \int_{y \in B_r(x) \cap P} s(y)$ is the aggregate strategy around x. This game models a situation where there are k congestible resources and a continuum of players arranged in space. Each player derives a certain value from using each resource, but their ability to use the resource diminishes with the amount of surrounding players who

[9] We discretize a 1D interval in the obvious way, i.e., by placing N equally-spaced points across the interval. We discretize a higher-dimensional cube by taking the first N points of the Roberts sequence (Roberts, 2018), a low-discrepancy (a.k.a. quasi-random) sequence (Kuipers and Niederreiter, 1974; Niederreiter, 1992) with good approximation properties. We discretize a simplex using a transformation of the low-discrepancy sequence for the cube (Pillards and Cools, 2005).

[10] More details, including how to sample states conditioned on observations, can be found in Bichler, Fichtl, Heidekrüger, et al. (2021).

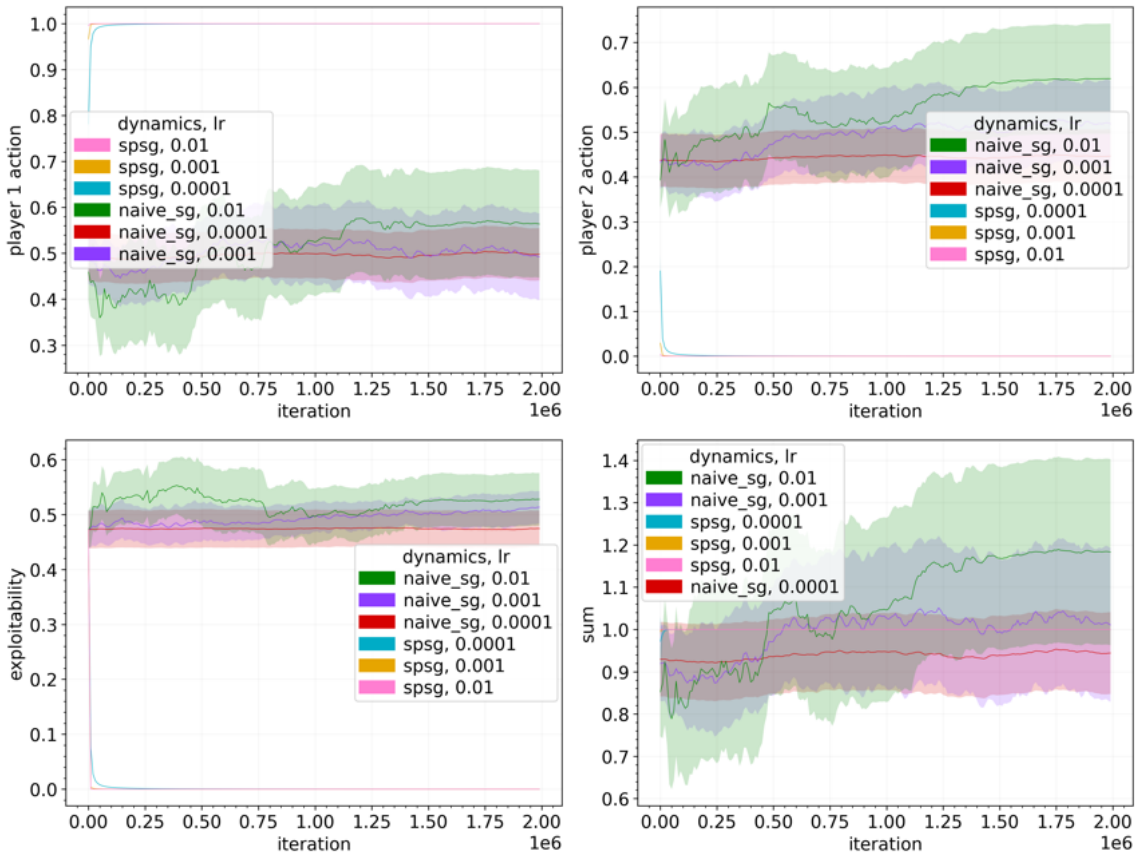


Figure 5.38: Sum game.

also choose that resource. Thus each player wants to satisfy a particular endogenous *bias*, but also wants to exogenously *anti-conform* or anti-coordinate with its neighbors. The game resembles the Lenz–Ising model (Lenz, 1920; Ising, 1925) of statistical mechanics, which models ferromagnetism and the magnetic dipole moments of atomic spins. In particular, each player's strategy corresponds to an **expected spin vector**, and the integral of the utility function over all players (i.e., the utilitarian social welfare) yields the negative Hamiltonian of that model. This so-called **Ising game** has been studied by Galam and Walliser (2010), Xin, Yang, and Huang (2017), Leonidov, Savvateev, and Semenov (2020), Leonidov, Savvateev, and Semenov (2024), and Feldman, Kim, and Palmer (2024), among others. In our experiments, we use $r = 0.1$, $k = 3$, and $b(x) = [x \in B_{1/4}(\mathbf{1}/2)]\mathbf{e}_0$. The latter incentivizes players near the center of the domain to choose 0. We do this to break spatial symmetry and eliminate a trivial everywhere-constant equilibrium where players mix equally between all resources. Exploitabilities are shown in Figure 5.39 and 5.40. In both cases, our method decreases exploitability across iterations. Learned strategy profiles are shown in Figure 5.39 and 5.40. As expected, players inside the ball at the center of the domain choose 0. Beyond this ball, players

alternate between the 3 choices (except immediately around the ball, where the players always skip choice 0).

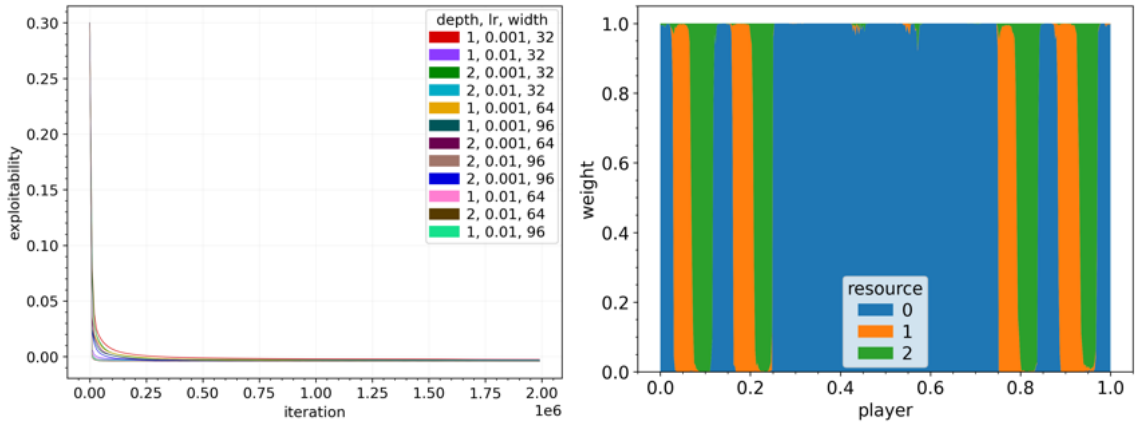


Figure 5.39: 1D anti-coordination game. Left: Exploitabilities. Right: A learned strategy profile. This is a stacked bar plot: each vertical slice shows a point on the simplex.

5.4.2.3 Discontinuous game

Let $\varepsilon, \delta > 0$. Let $P = [0,1]$, $S(p) = [0,1]$, and $u(s,p) = \int_{q \in B_\varepsilon(p) \cap P} [|s(p) - s(q)| < \delta]$ for $p \in [0.1, 0.4] \cup [0.6, 0.9]$, $s(p)$ for $p \in [0, 0.1) \cup (0.9, 1]$, and $-s(p)$ for $p \in (0.4, 0.6)$. Each player in the first region has an incentive to choose an action within δ of its neighbors', making this a type of conformity game. The integrand is discontinuous. Consequently, taking the gradient of the integrand yields an incorrect, biased estimator of the gradient of the integral. We overcome this obstacle by replacing the gradient used inside the SPSG with a pseudo-gradient, as described in Section 2.8. In our experiments, we use $\varepsilon = \delta = 0.1$. Exploitabilities are shown in Figure 5.41. In the legend, "smooth_scale" denotes the smoothing scale σ used for the pseudo-gradient, with $\sigma = 0$ reserved for no smoothing. As expected, smoothing is needed to learn a strategy profile with low exploitability.

5.4.2.4 Circle game

Let S^1 be the unit circle. Let $r \in (0, \frac{1}{2})$ be an interaction radius. Let $P = S^1$, $S(p) = S^1$, and $u(s,p) = 2 \int_{q \in [p-r,p]} d(s(p), s(q)) - \int_{q \in [p,p+r]} d(s(p), s(q))$. Here, $d(x,y) = \min\{|x-y|, 1-|x-y|\}$ denotes the distance on the unit circle (with wraparound). The utility contains a repulsion term and an attraction term. Each player wants to be similar to the players ahead of it, but more dissimilar to the players behind it. Around a circle, that creates a chasing effect: everyone wants to pull away from those just behind, which in turn makes those behind want to pull away from someone else, and so on. With stronger repulsion than attraction, this chase can never close into a consistent profile. Thus this game has no pure-strategy NE.[11] Therefore, players need to be able to randomize over the continuous action space. To do this—i.e., model mixed strategies over the infinite action space—we

[11] It has an MSNE where each player mixes uniformly on S^1.

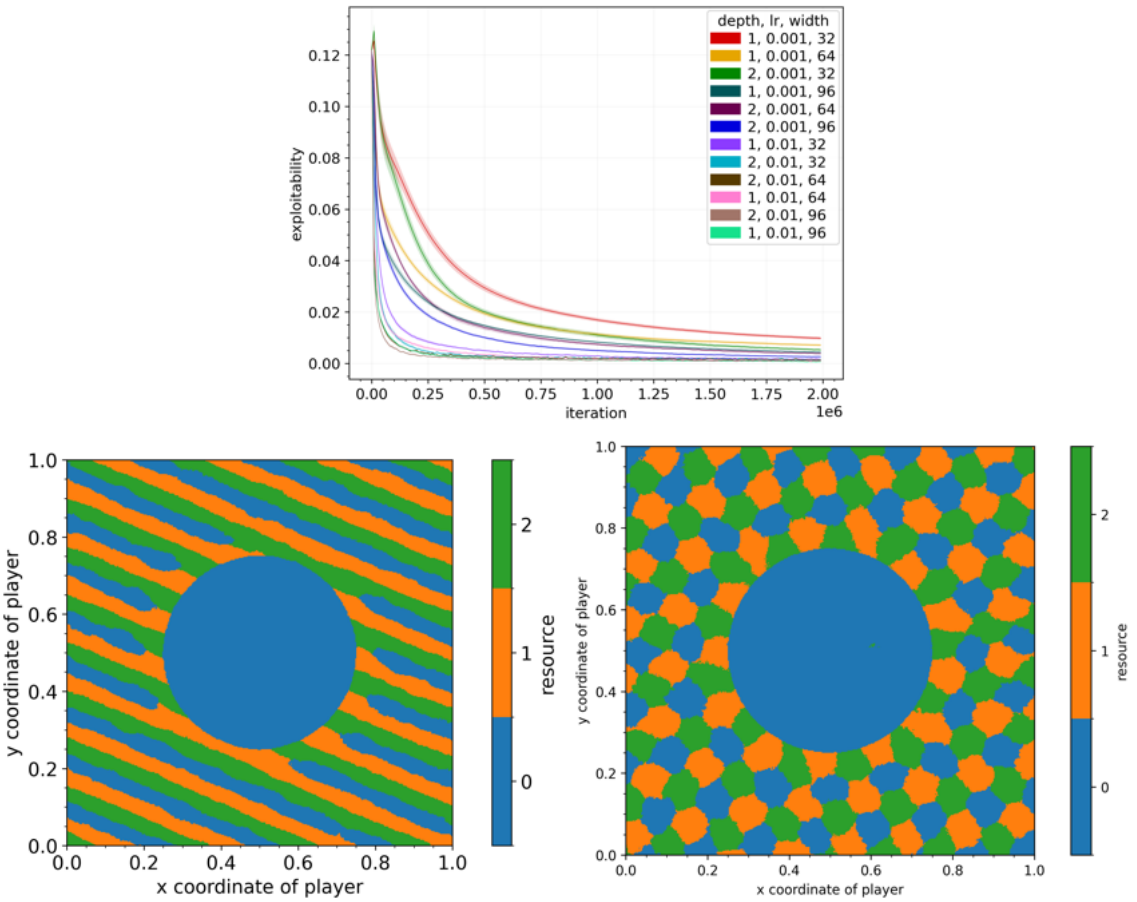


Figure 5.40: 2D anti-coordination game. Top: Exploitabilities. Bottom: Two learned strategy profiles. This is a categorical plot: each point shows the player's top choice.

use the randomized policy network approach described by Martin and Sandholm (2023), which injects latent noise into the input of the network to induce a randomized output. In our experiments, we use $r = 0.1$. Exploitabilities are shown in Figure 5.42. In the legend, "noise dim" denotes the dimensionality of the Gaussian noise passed as additional input to the network. Learned strategy profiles are shown in Figure 5.43. As expected, noise is needed to learn a strategy profile that can randomize.

5.4.2.5 Cournot game

Let $d \in \mathbb{N}$ be a spatial dimension. Let $r \in \mathbb{R}$ be an interaction radius. The player set is $P = [0,1]^d$. The strategy set function is $S(x) = [0,1]$. The utility function is $u(s,x) = s(x)P(\bar{s}(x)) - C(x,s(x))$, where $\bar{s}(x) = \int_{y \in B_r(x) \cap P} s(y)$ is the aggregate strategy around x. This game models a spatial

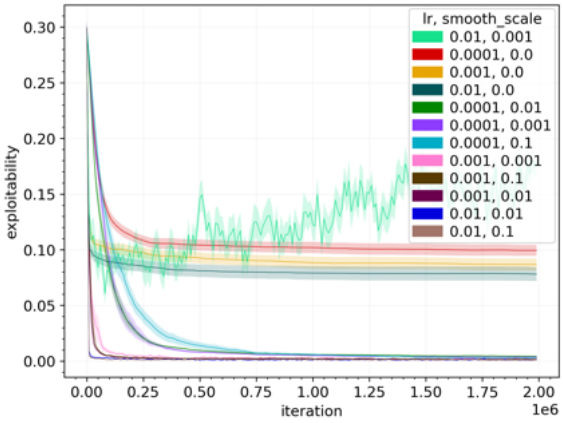


Figure 5.41: Discontinuous game.

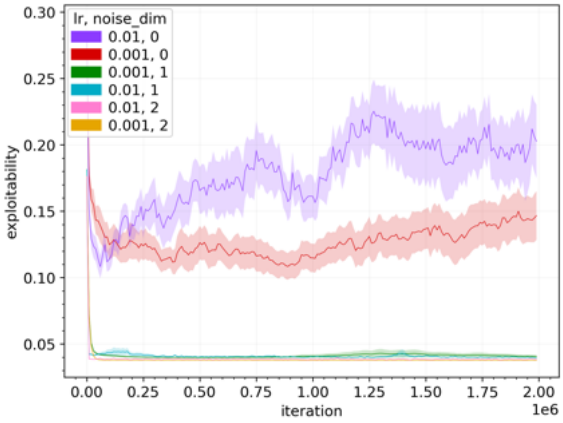


Figure 5.42: Circle game.

Cournot competition. Cournot competition is a classic economic model dating back to Cournot (1838) and Cournot (1863) in which firms compete in a market for a homogeneous good by choosing a quantity of output to produce independently and simultaneously. Versions with a continuum of players have been studied extensively (e.g., Novshek (1985) and Chan and Sircar (2015)). Each player represents a *firm* choosing what quantity, if any, to produce of a good. A firm's profit is revenue $-$ cost $= qP(\bar{q}) - C(q)$, where q is its output, $\bar{q}$ is the aggregate output of its neighborhood, P is the *inverse demand function* (which yields the market price associated with an aggregate output), and C is the firm's *production cost function*. For simplicity, we use a linear inverse demand function $P(\bar{q}) = a - b\bar{q}$ and linear production cost function $C(q) = cq$. Thus, c represents a *marginal cost of production*. In our experiments, we use $r = 0.1$, $a = b = 1$, and $c = 0.5$. Exploitabilities are shown in Figure 5.44 and Figure 5.45. In both cases, our method decreases exploitability across iterations. Learned strategy profiles are shown in Figure 5.44 and Figure 5.45. At the edges of the

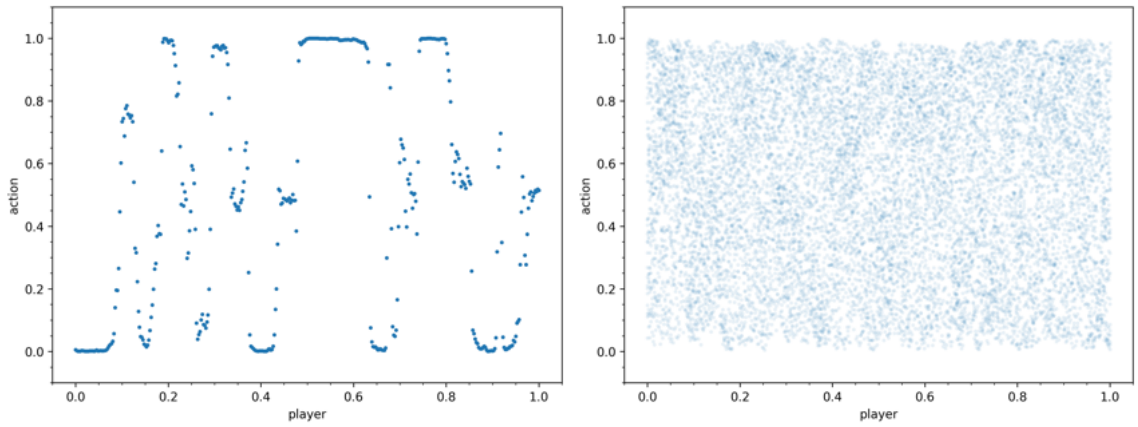


Figure 5.43: Learned strategy profiles for the circle game. Left: Noise dimension 0 (no noise). Right: Noise dimension 1.

domain, players always choose to produce. This is because they are surrounded by fewer players in total, and thus face less competition. Meanwhile, players in the interior of the domain alternate between producing or not in narrow horizontal stripes.

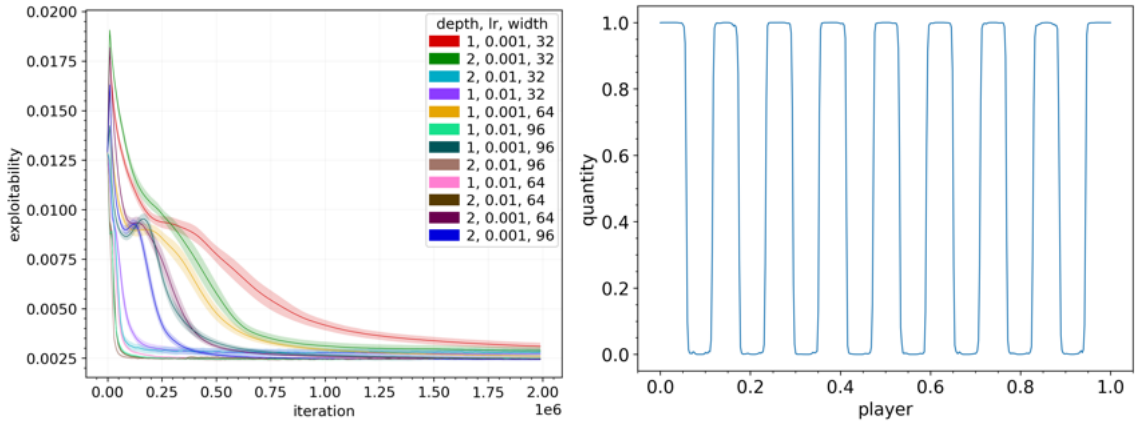


Figure 5.44: 1D Cournot game. Left: Exploitabilities. Right: A learned strategy profile.

5.4.2.6 Bayesian Cournot game

We now consider a variant of the Cournot game with **incomplete information**. Specifically, we give each player only limited information about the marginal cost c. For this game, we let the state be the marginal cost c, the state space be the unit interval $[0, 1]$, the initial state distribution be

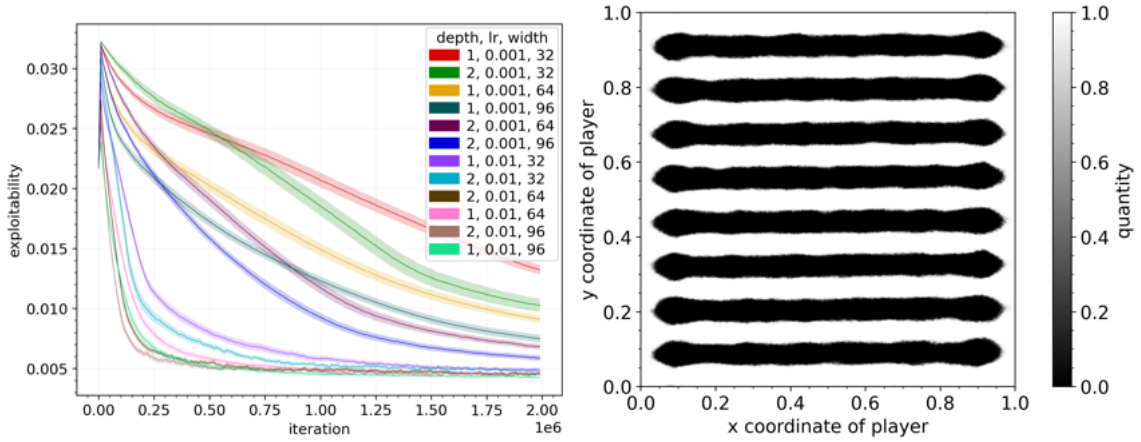


Figure 5.45: 2D Cournot game. Left: Exploitabilities. Right: A learned strategy profile.

the standard uniform distribution, the observation space be $\mathbb{R}$, and the observation function for player i yield $\max\{c, \|i\|_2\}$ deterministically. The actions and utility function are as before. This game gives players closer to the origin more information. Exploitabilities are shown in Figure 5.46 for $d = 2$. Our method decreases exploitability across iterations. A learned strategy profile is shown in Figure 5.46. As expected, players near the origin can make a better decision, because they have more information about the true state of the world. For example, players near the origin decrease production when the marginal cost is high.

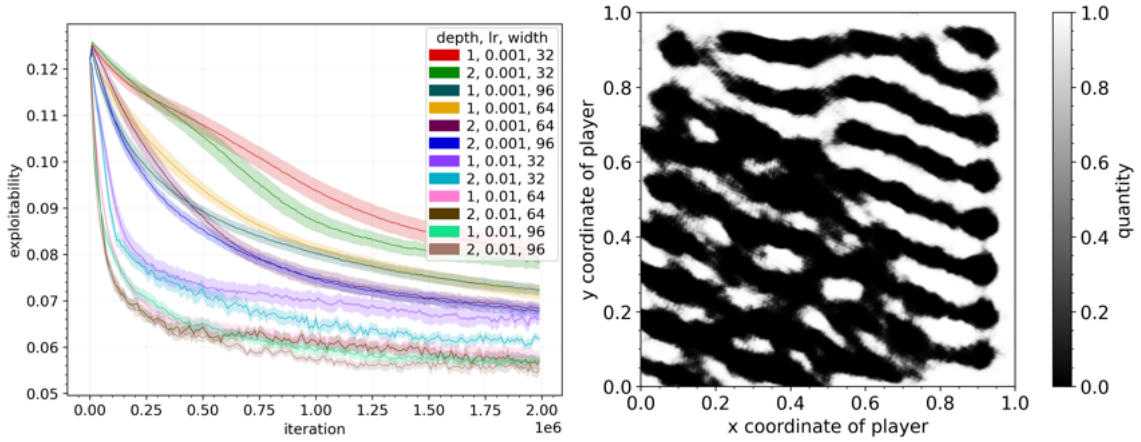


Figure 5.46: 2D Bayesian Cournot game. Left: Exploitabilities. Right: A learned strategy profile, under a state (marginal cost) of 0.8.

5.4.2.7 Quadratic-cost Cournot game

We replace the linear cost function $C(q) = cq$, with a *quadratic* cost function $C(q) = cq^2$. We keep the same $c = 0.5$. Results are shown in Figure 5.47 and 5.48. Unlike the original Cournot game, this time, most players choose outputs strictly between 0 and 1, with players in the interior of the domain choosing values near 0.5. Players near the edges of the domain are, as before, more likely to choose high outputs (of around 0.7 to 0.8), because they face less competition.

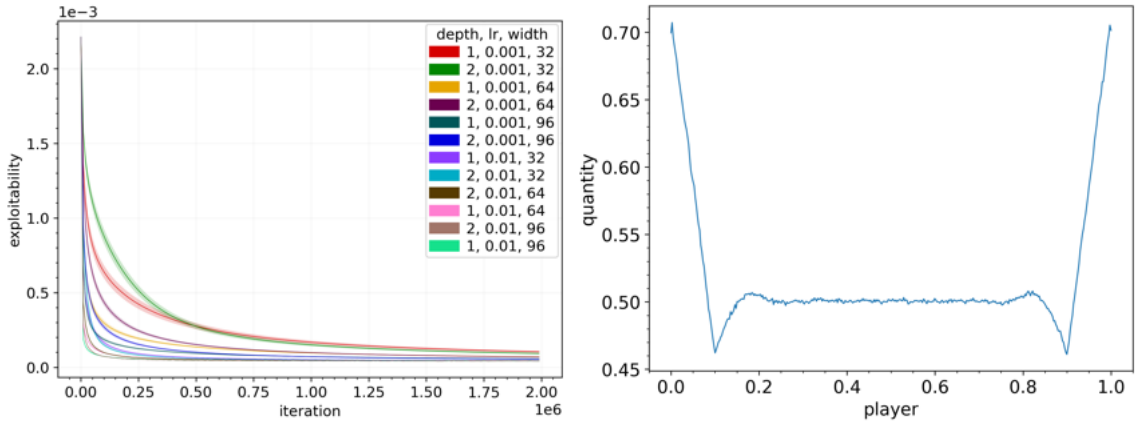


Figure 5.47: 1D quadratic-cost Cournot game. Left: Exploitabilities. Right: A learned strategy profile.

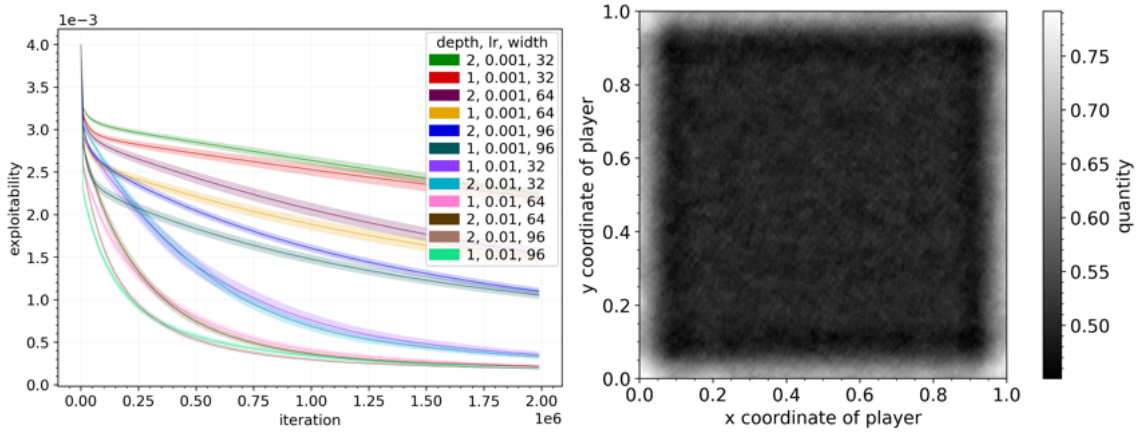


Figure 5.48: 2D quadratic-cost Cournot game. Left: Exploitabilities. Right: A learned strategy profile.

5.4.2.8 Heterogeneous-cost Cournot game

We now consider another variant of the spatial Cournot game. This time, we let the marginal cost c vary across players. Specifically, we let $c(x, y) = \sin(\pi x)^2 \sin(\pi y)^2 + \frac{1}{2} \sin(3\pi x)^2 \sin(3\pi y)^2$. This is shown in Figure 5.49. Exploitabilities are shown in Figure 5.50. A learned strategy profile is shown in Figure 5.50. The strategy profile is heterogeneous in a way that reflects the underlying marginal cost field.

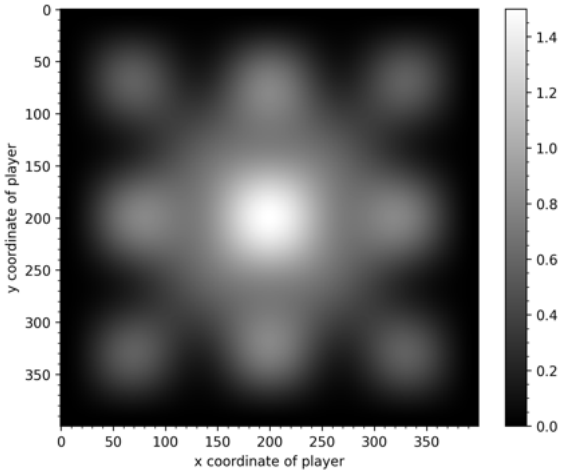


Figure 5.49: Marginal costs for heterogeneous-cost Cournot game.

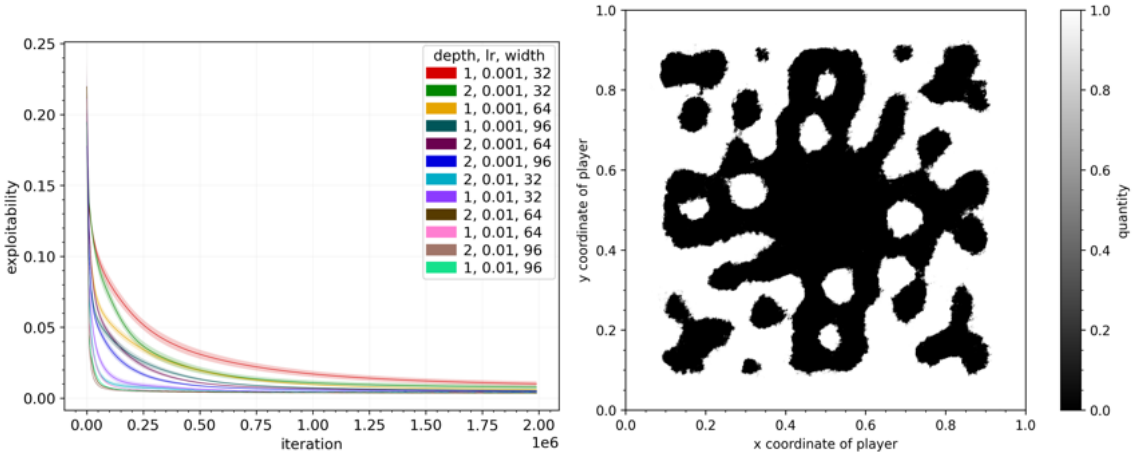


Figure 5.50: Heterogeneous-cost Cournot game. Left: Exploitabilities. Right: A learned strategy profile.

5.4.2.9 Conformity game

Let $d \in \mathbb{N}$ be a spatial dimension. Let $r \in \mathbb{R}$ be an interaction radius. The player set is $P = [0,1]^d$. The strategy set function is $S(x) = [0,1]$. The utility function is

$$u(s,x) = -\begin{cases} (s(x)-0)^2 & \max(x) < 0.3 \\ (s(x)-1)^2 & \min(x) > 0.7 \\ (s(x)-\bar{s}(x))^2 & \text{otherwise} \end{cases} \tag{5.28}$$

where $\bar{s}(x) = \int_{y \in B_r(x) \cap P} s(y)$ is the aggregate strategy around x. This models a situation where some players (leaders) want to conform to a fixed target, while other players (followers) want to conform to their neighbors. We let $r = 0.1$ in our experiment. Experimental results are shown in Figure 5.51 and Figure 5.52. As expected, leaders choose their target values, while followers interpolate between their neighbors' values.

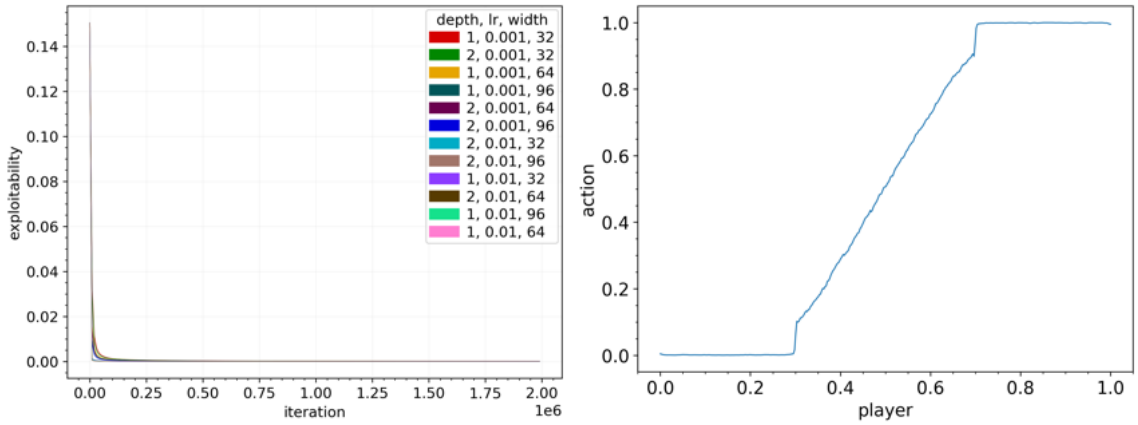


Figure 5.51: 1D conformity game. Left: Exploitabilities. Right: A learned strategy profile.

5.4.2.10 Permutation game

The **Plackett–Luce (PL) model** (Luce, 1959; Plackett, 1975) is a probability model for *rankings*, that is, permutations of a set of items. It is based on Luce's axiom of choice (Luce, 1959; Luce, 1977), which states that the probability of choosing one item over another does not depend on the set of items from which the choice is made. Under this model, each item i has a score $x_i \in \mathbb{R}$, and the probability of choosing item i from a set of items S is $\frac{\exp x_i}{\sum_{j \in S} \exp x_j} = \text{softmax}(\mathbf{x})_j$. A ranking is modeled as a sequence of choices. At each point in the sequence, an item is chosen from the remaining set of items according to the aforementioned probability. There is a fast, efficient way to sample from the PL model, described in Yellott (1977), Grover et al. (2019, Proposition 5), and Gadetsky et al. (2020, Lemma 1). Let $n \in \mathbb{N}$ be the number of items. For each item $i \in [n]$, let $x_i \in \mathbb{R}$ be its score and $g_i \sim$ Gumbel be noise sampled from the standard Gumbel distribution. Let the output permutation be $\pi = \text{argsort}(\mathbf{x} + \mathbf{g})$, where argsort yields the indices that sort its

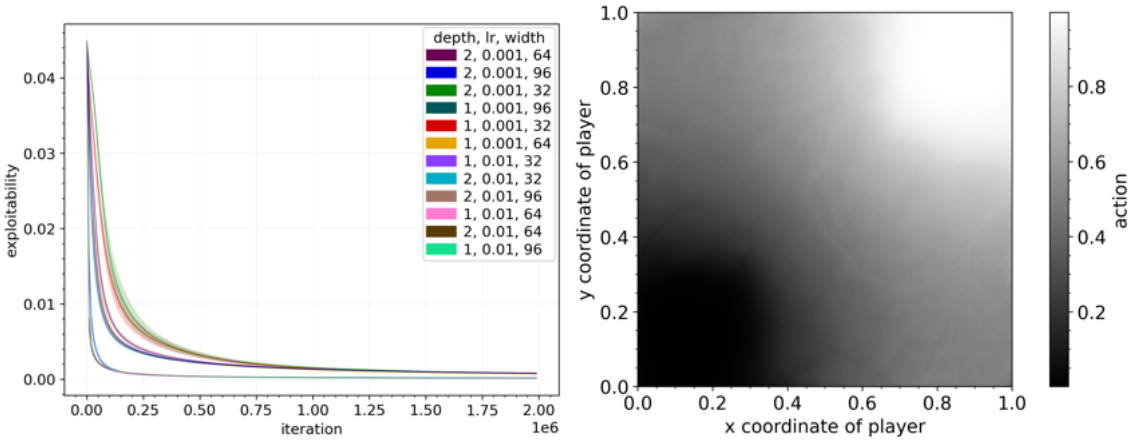


Figure 5.52: 2D conformity game. Left: Exploitabilities. Right: A learned strategy profile.

input in descending order. This is sometimes called the *Gumbel argsort trick*. We denote the set of permutations of $[n]$ by $n!$, so $\pi \in n!$ is an element of this set.

Let $d, n \in \mathbb{N}$. Let $P = [0,1]^d$ be the set of players. For each player $p \in P$, let $S(p) = \triangle[n]$ be the set of strategies. Given a strategy profile s, for each player $p \in P$, let $\pi(p) \sim \mathrm{PL}(\{\log s(p)_i\}_{i\in[n]})$ be a permutation sampled from its strategy according to the PL model described above (with the samples being independent across players). In other words, each player independently chooses a permutation of $[n]$. Therefore, this is a game with a combinatorial action space. Let the utility function, expressed in terms of the sampled permutations, be

$$u(\pi, p) = \begin{cases} [\pi(p) = \mathrm{id}_{[n]}] & \|p - \frac{\mathbf{1}}{2}\|_2 < \delta \\ \int_{q \in B_r(p)} f(\pi(p), \pi(q)) & \text{otherwise} \end{cases} \tag{5.29}$$

Here, $\delta, r \in \mathbb{R}_{\geq 0}$ are distances, $\mathrm{id}_{[n]} = \{i\}_{i\in[n]}$ is the identity permutation, and $f : n! \times n! \to \mathbb{R}$ is a pairwise interaction function. This incentivizes players inside a δ-ball at the center of the domain to choose the identity permutation. Let the interaction function be as follows:

$$f(\pi, \sigma) = \sum_{i\in[n]} [\![\pi_i > \sigma_i]\!] - [\![\pi_i < \sigma_i]\!] \tag{5.30}$$

That is, a player's payoff is the number of positions where it beats the opponent, minus the number of positions where it loses to the opponent. This induces a non-transitive dominance relation between permutations. Every permutation of length ≥ 3 is strictly dominated by another distinct permutation.

We run experiments with $r = 0.1$, $\delta = 0.2$, and $d = 2$. Since this game has a discontinuous utility function, we use pseudo-gradients. Exploitabilities are shown in Figure 5.53. A learned strategy profile is shown in Figure 5.54.

In Section C.2, we compare our method to the naive approach of discretizing the player space, showing superior performance.

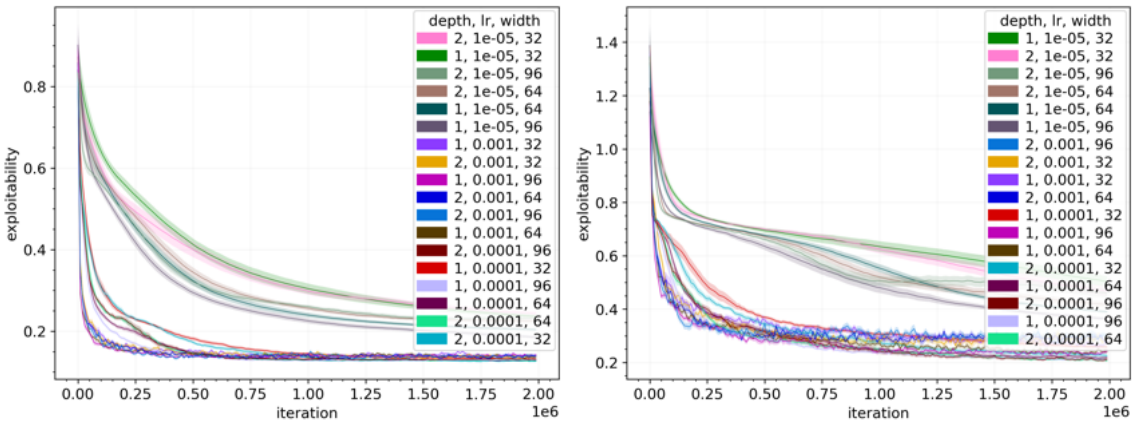


Figure 5.53: Permutation game with $n = 3$ and $n = 4$.

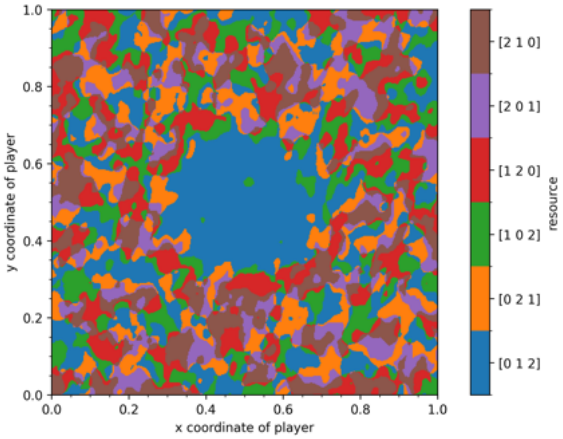


Figure 5.54: Strategy profile for permutation game with $n = 3$. This is a categorical plot. Each point shows the player's top choice.

Chapter 6

Proposed work

In this section, we describe the next steps for our project as well as proposed future work. We aim to combine all of the techniques we have presented so far, to create a unified method that can solve (infinite-state, infinite-action, infinite-player, etc.) games in full generality. These results are obtainable within a reasonable amount of time. We want to emphasize that the methodological core is largely complete. The remaining work is primarily empirical and expository.

6.1 Methods

Our first step is to simply **integrate and unify** all of the techniques we have presented so far. We do this as follows:

- Use a P2SN to represent strategy profiles.
- Use SPSG for the simultaneous gradient analogue in this shared-parameter setting.
- Use the infinite-player analogue of JPSPG for efficient estimation of this SPSG.
- Use ApproxED-BRF for the learning dynamics.

Further details and extensions are given below. We will test these on the proposed benchmarks, which are described in Section 6.2.

6.1.1 Extension to multi-step games

Our experiments with P2SN in the original paper (Section 5.4.1) were limited to single-step normal-form games and Bayesian games. However, we can straightforwardly generalize our method to apply to multi-step games. To do this, we can simply apply pseudo-gradients (Section 2.8) to the stochastic function that maps the parameters of the P2SN to sampled episode returns for a given player.

In the multi-step case, we can no longer estimate best response values as we did for Bayesian games, that is, via the discretization we described at the end of Section 3.3. However, we can estimate best response values by running single-agent RL for the given player, treating the other players as a fixed part of the environment. To do this, we can use pseudo-gradients like we do in the multi-player case, but with just 1 player. This is essentially the technique used by Salimans et al. (2017). In particular, we block propagation of gradients through P2SN queries for other players. This is known

as a "stop gradient" operation in automatic differentiation. It is the same operation we use to block gradients for other players when estimating the SPSG.

6.1.2 Extension to infinite-step games

Our multi-step extension (Section 6.1.1) applies to games with a finite number of discrete timesteps. We propose to extend the framework further to the **infinite-step** regime—specifically, continuous-time **differential games** (Isaacs, 1965; Friedman, 1971a).

In differential games, each player's action at any point in time is a *rate* (velocity, thrust, bid intensity). For example, in a game where players control positions, their outputs could be a velocity (derivative of position with respect to time) instead of a displacement (change in position). Some of our proposed benchmarks, such as Orbit, Traffic, and Foraging, are naturally continuous-time, and treating them as genuine differential games (rather than coarse-grained discrete approximations) requires this extension.

Differential games combine optimal control theory with game theory to model continuous-time conflicts. In these dynamic models, the state of the system evolves over time according to a set of (potentially stochastic) ODEs. The field was pioneered by Rufus Isaacs in the 1950s and formally established in his seminal text (Isaacs, 1965). A classic illustrative problem is the **homicidal chauffeur problem** (Isaacs, 1965; Merz, 1974; Patsko and Turova, 2001), which models a pursuit-evasion scenario between a fast but unmaneuverable pursuer and a slow but agile evader. Today, differential games are widely applied to solve complex continuous-time problems in economics, military strategy, and engineering (Dockner et al., 2000).

6.1.2.1 Continuous-time POSG formulation

The POSG formulation of Section 3.5 generalizes to continuous time by replacing the discrete transition kernel T with a general **jump-diffusion stochastic ODE**:

$$\mathrm{d}s_t = \mu(s_t, a_t)\,\mathrm{d}t + \sigma(s_t, a_t)\,\mathrm{d}W_t + \int_{z \sim N(\mathrm{d}t)} h(s_t, a_t, z) \tag{6.1}$$

whose components are as follows:

- t is the time index.
- s_t is the instantaneous state.
- a_t is the instantaneous joint action.
- W_t is a standard Wiener process (Wiener, 1923) (i.e., Brownian motion), representing continuous, normally distributed noise.
- z is a jump size for the state.
- $N(\mathrm{d}t)$ is a Poisson random measure (Poisson, 1837; Wiener, 1938), which is the distribution of jump sizes that occur in the infinitesimal time interval $\mathrm{d}t$.
- μ is the drift coefficient, representing the deterministic rate of change.
- σ is the diffusion coefficient (or volatility), scaling the continuous random noise.
- h is the jump amplitude function.

The return is the Itô integral (Itô, 1944) of the instantaneous rewards r_t weighted by the (possibly matrix-valued) discounts γ_t.

6.1.2.2 Proposed approach

The existing components of our framework generalize as follows.

1. **Trajectory rollout**: Episodes are sampled by numerically integrating the SDE over the time horizon. In smooth regions of the state space, we use the Dormand–Prince adaptive-stepsize method (Dormand and Prince, 1980; Hairer, Nørsett, and Wanner, 1993), which automatically shrinks the stepsize to maintain a target truncation error. Near optimal switching surfaces—points at which a player's best action changes discontinuously—we use event detection to locate the switch time precisely and restart the integrator on either side, preventing the solver from smearing the discontinuity. We will use JAX-native ODE/SDE solvers like Diffrax (Kidger, 2021), so that the entire rollout remains a pure function of the P2SN parameters and a PRNG (pseudo-random number generator) key, consistent with our existing JPSPG implementation.

2. **Parameter updates**: We apply the JPSPG estimator (Section 5.3.1) to the composition (P2SN parameters) $\to$ (integrated trajectory) $\to$ (episode return). This sidesteps the two standard obstacles to differentiating through an SDE solver: discontinuities at switching surfaces, where the adjoint method fails without careful treatment, and the memory cost of reverse-mode differentiation over long horizons. The constant per-iteration evaluation count of JPSPG in the number of players is essential here, since continuous-time rollouts are individually expensive.

3. **Best-response estimation**: Exploitability is estimated, as in Section 6.1.1, by training a single-agent best response against the frozen population policy, which is now a continuous-time stochastic optimal control problem. If the environment is smooth over time, we can train a continuous-time actor against the frozen opponents' P2SN queries. If it is not (such as in discrete collision events), we fall back to single-player JPSPG on the rollout return.

6.1.2.3 Open questions

These require experimental resolution:

1. **Variance–tolerance coupling**: JPSPG's variance scales with the perturbation radius σ. Dormand–Prince's local truncation error scales with the step tolerance ε. The interaction of the two has not been studied, to our knowledge. We will characterize it empirically and, if needed, couple the two hyperparameters.

2. **Convergence criterion**: Unlike the discrete-time case, exploitability is itself estimated via an inner optimization that never terminates cleanly. We will report exploitability trajectories alongside an explicit compute budget, following the convention of our earlier work (Section 5).

3. **Failure mode**: If the approach is infeasible on a benchmark (e.g., if best-response training does not converge within budget), we fall back to a fine-grained discrete approximation and report the gap.

6.1.2.4 Prior work

In a differential game, players continuously make decisions to optimize their individual payoff functions over a specified time horizon. A well-known category is the **two-player zero-sum differential**

game, where one player's gain is exactly the other player's loss (Başar and Olsder, 1999). The fundamental mathematical tool for solving such zero-sum games is the **Hamilton–Jacobi–Isaacs** (HJI) equation. The continuous-time HJI equation is

$$\min_u \max_v \left[\nabla V \cdot f(x,u,v) + L(x,u,v)\right] = 0 \tag{6.2}$$

where V is the value function, u and v are the control inputs, and f represents the system dynamics (Friedman, 1971b). The HJI equation characterizes equilibrium only in the two-player zero-sum case. For general-sum and many-player settings, our success metric remains empirical exploitability, not HJI residual.

Solving differential games analytically is rarely feasible, requiring **numerical integration techniques** to compute the optimal trajectories and value functions (Bardi and Capuzzo-Dolcetta, 1997). One approach transforms the PDE into a system of ODEs, using the method of characteristics. These characteristic ODEs form a two-point boundary value problem that can be solved using iterative numerical integration methods like shooting methods or multiple shooting. Standard explicit methods, such as the classic fixed-step fourth-order Runge–Kutta method, often struggle with the discontinuities and switching surfaces inherent in optimal control problems. To efficiently resolve these sharp gradients and optimal switching times, practitioners employ **adaptive stepsize algorithms**, such as the Runge–Kutta–Fehlberg or Dormand–Prince methods (Hairer, Nørsett, and Wanner, 1993). These adaptive integrators compute two simultaneous approximations of different orders to estimate the local truncation error at each time step. If the estimated error exceeds a predefined tolerance, the solver automatically reduces the stepsize to maintain accuracy, whereas it increases the stepsize in smooth regions to conserve computational resources.

6.1.3 Using ApproxED-BRF to train the P2SN

Our original P2SN paper uses SPSG (Section 5.4.1), the analogue of simultaneous gradient ascent (Section 4.3) for the shared-parameter case, to evolve the parameters in time. However, as we suggested in the ApproxED paper (Section 5.2.1.1), we could use an alternative parameter dynamics based on **minimization of a local approximation of exploitability**. Specifically, we could use ApproxED-BRF. This could lead to convergence to equilibrium in a wider range of cases than the simultaneous gradient ascent dynamics, as we observed in the ApproxED paper.

Now, in our case, the strategy profile is represented by a P2SN. Therefore, the BRF has to map this P2SN to another P2SN. Since the strategy profile is itself a neural network, this means the BRF must be a **hypernetwork** (Schmidhuber, 1992; Ha, Dai, and Le, 2017; Lorraine and Duvenaud, 2018; MacKay et al., 2019; Bae and Grosse, 2020), that is, a neural network that maps one neural network to another. In our ApproxED paper, we already used hypernetworks for strategy profiles in which the strategies were policy networks.

Recall from Section 5.4.1 that the shared-parameter simultaneous gradient (SPSG) is defined as

$$\left[\frac{\mathrm{d}}{\mathrm{d}\phi}\int_{i\sim\mu} u(s_\theta[i \mapsto (s_\phi)_i])_i\right]_{\phi=\theta} \tag{6.3}$$

Recall from Section 5.2.1.1 that the BRF evolves as follows:

$$\dot{x} = -\nabla_x \mathrm{NI}(x, b_\theta(x)) \tag{6.4}$$

$$\dot{\theta} = +\nabla_\theta \mathrm{NI}(x, b_\theta(x)) \tag{6.5}$$

If we let θ refer to the parameters of the P2SN and ϕ refer to the parameters of the best-response hypernetwork (which takes the P2SN's parameters as input), we have

$$\dot{\theta} = -\nabla_\theta \int_{i \sim \mu} (u(s_\theta[i \mapsto b_\phi(\theta)_i])_i - u(s_\theta)_i) \tag{6.6}$$

$$\dot{\phi} = +\nabla_\phi \int_{i \sim \mu} (u(s_\theta[i \mapsto b_\phi(\theta)_i])_i - u(s_\theta)_i) \tag{6.7}$$

6.1.4 Mitigating catastrophic forgetting in ApproxED-BRF

Recall that ApproxED-BRF (Section 5.2.1.1) trains a best response function against the current strategy profile. The motivation is that a function can model best responses for many strategy profiles simultaneously, and thus in principle can "retain" memory of what good responses are for prior strategy profiles, avoiding the cycling problems encountered by naive learning, that is, simultaneous gradient ascent (Section 4.3). However, since the BRF trains only against the current strategy profile on each iteration, it is in principle still possible for the BRF to "forget" what it has learned in the distant past, i.e., how to best respond to strategy profiles that were only encountered in the distant past.

In the ML literature, **catastrophic forgetting** (or catastrophic interference) is a fundamental limitation in artificial neural networks where the model completely and abruptly forgets previously learned information upon learning new data or tasks (McCloskey and Cohen, 1989). This occurs because the weights (parameters) of the network that were optimized for a previous task are heavily altered through backpropagation to minimize the error on the new task. As a result, the network's performance on the old task drastically degrades (French, 1999). This prevents standard neural networks from achieving "continual" or "lifelong" learning, a key feature of human biological intelligence.

Researchers have proposed various strategies to mitigate catastrophic forgetting, which generally fall into three main categories.

Regularization-based methods. These methods introduce an additional term to the loss function to penalize changes to weights that are deemed critical for previously learned tasks.

- **Elastic weight consolidation (EWC):** EWC slows down learning on certain weights based on how important they are to previously seen tasks, estimating this importance using the Fisher information matrix (Kirkpatrick et al., 2017).
- **Learning without forgetting (LwF):** This approach utilizes knowledge distillation. It uses the model's own prior outputs on new task data to create a regularization target, ensuring the network's responses to old tasks do not drift significantly (Li and Hoiem, 2017).

Architectural and parameter isolation methods. These methods dynamically alter the architecture of the neural network to dedicate distinct subsets of parameters to different tasks.

- **Progressive neural networks:** This method instantiates a new neural network column for each new task. It prevents forgetting by freezing the weights of previous columns and only allowing lateral connections to the new column (Rusu et al., 2016).

Rehearsal and replay methods. Rehearsal techniques prevent forgetting by periodically retraining the network on a subset of the old data while learning the new data.

- **Experience replay:** A memory buffer stores exact examples from previous tasks, which are interleaved with new data during training (Rebuffi et al., 2017).
- **Generative replay:** Instead of storing raw data, a separate generative model (like a GAN) is trained to generate pseudo-data from past distributions, which is then replayed to the main network (Shin et al., 2017).

We opt for the replay approach. That is, instead of relying only on the implicit flexibility of the BRF, we go one step further and *explicitly* train the BRF against multiple strategy profiles on each iteration. Specifically, instead of training against only the current strategy profile (i.e., the P2SN's parameters), we maintain an **experience replay buffer** of past strategy profiles.

We can fill the replay buffer via **reservoir sampling** (Vitter, 1985). Reservoir sampling is a family of randomized algorithms used to select a simple random sample of k items from a population of unknown or vast size n, where n is either too large to fit into memory or is a continuous data stream. It allows one to maintain a representative sample at any point during the stream.

By utilizing reservoir sampling to manage the experience replay buffer, continual learning models achieve the following benefits.

- **Uniform historical representation:** Reservoir sampling mathematically guarantees that at any given time step n, every single data point seen so far has an exact equal probability (k/n) of being in the buffer. This ensures the replay buffer remains an unbiased, uniform sample of the entire data distribution seen over the model's lifetime.
- **Memory efficiency:** The model does not need to store the entire dataset or even know the total size of the dataset in advance. It only requires the k slots in the buffer and a counter for n, making it highly efficient for continuous learning environments.

In summary, we store past strategy profiles at random, and on every iteration, we sample one and train the BRF against it. This makes the BRF less likely to overfit to the current strategy profile (or recent profiles), and mitigates catastrophic forgetting.

6.1.5 Modifying BRF's update scheme to use other learning dynamics

BRF (Section 5.2.1.1) resembles simultaneous gradient ascent (Section 4.3) applied at the level of strategy profiles rather than individual strategies. Specifically, recall from Section 4.3 that simultaneous gradient ascent is as follows for a two-player zero-sum game:

$$\dot{x} = -\nabla_x u(x, y) \tag{6.8}$$

$$\dot{y} = +\nabla_y u(x, y) \tag{6.9}$$

Recall from Section 5.2.1.1 that the BRF evolves as follows:

$$\dot{x} = -\nabla_x \mathrm{NI}(x, b_\theta(x)) \tag{6.10}$$

$$\dot{\theta} = +\nabla_\theta \mathrm{NI}(x, b_\theta(x)) \tag{6.11}$$

This is because the goal is to find

$$x^* \in \operatorname*{argmin}_{x \in \mathcal{X}} \mathrm{NI}(x, b(x)) \tag{6.12}$$

such that $b : \mathcal{X} \to \mathcal{Y}$ is a function that satisfies

$$b(x) \in \operatorname*{argmax}_{y \in \mathcal{Y}} \mathrm{NI}(x, y) \tag{6.13}$$

In Section 4.3, we saw that there are other learning dynamics that modify the simple simultaneous gradient ascent scheme. These include extragradient and optimistic gradient, as well as other methods that involve higher-order derivatives. We could substitute these for the simultaneous gradient ascent that BRF uses at the level of strategy profiles (versus the BRF's parameters).

Since the game that ApproxED-BRF is trying to solve is technically a Stackelberg game (we are trying to find the min-max specifically, not the max-min, so the order matters), we can also consider dynamics made specifically for **Stackelberg games**. Examples of these can be found in Fiez, Chasnov, and Ratliff (2019), Fiez, Chasnov, and Ratliff (2020), Goktas and Greenwald (2021), Goktas and Greenwald (2022b), Goktas, Zhao, and Greenwald (2022b), and Goktas, Zhao, and Greenwald (2022a).

6.1.6 Joint-perturbation estimator for other learning dynamics

For this objective, the goal is to develop analogues of the JPSPG estimator (Section 5.3.1) for other learning dynamics (Section 4.3), including ones that involve higher-order derivatives and Hessian-vector products, so that these can be used efficiently as well.

Recall from Section 4.3 that the simultaneous gradient is

$$g_i = \frac{\mathrm{d}u(x)_i}{\mathrm{d}x_i} \tag{6.14}$$

Recall from Section 5.3.1 that the simultaneous pseudo-gradient estimator is

$$\hat{g}_i = \tfrac{1}{\sigma} u(x[i \mapsto x_i + \sigma z_i])_i z_i \tag{6.15}$$

Recall from Section 5.3.1 that the joint-perturbation JPSPG estimator is

$$\hat{g} = \tfrac{1}{\sigma} u(x + \sigma z) \odot z \tag{6.16}$$

This estimator can be re-used for both extragradient and optimistic gradient, which are comprised simply of expressions involving the simultaneous gradient. For the dynamics involving higher-order derivatives, the answer is less trivial. For these, the formulation of each dynamics in terms of the stop-gradient operator, as documented in Letcher (2018), is helpful. The expression inside each stop-gradient does not undergo perturbation for the derivative that is being computed via perturbation. For illustration, we consider a few of the dynamics. The same process can be carried out for the other dynamics in an analogous way.

To construct the updates elegantly, we define two base estimators using a single joint perturbation evaluation. The first-order pseudo-gradient for player i uses the antithetic central difference:

$$\hat{g}_i = \frac{u(x + \sigma z)_i - u(x - \sigma z)_i}{2\sigma} z_i \tag{6.17}$$

The scalar factor $\hat{d}_i$ captures the **second-order curvature** of the utility function along the specific perturbation vector, via a second-order central difference stencil:

$$\hat{d}_i = \frac{u(x + \sigma z)_i + u(x - \sigma z)_i - 2u(x)_i}{\sigma^2} \tag{6.18}$$

By leveraging the outer product of the perturbations, the cross-agent Jacobian blocks simplify to $\hat{J}_{ij} = \hat{d}_i z_i z_j^T$. This allows us to collapse the complex matrix operations of LOLA, CO, and SGA into isolated scalar products multiplied by the base perturbation vector z_i.

For the continuous-time differential equations, we define the exact individual gradient as $g_i \triangleq \nabla_{x_i} u(x)_i$.

Simultaneous gradient ascent simply evolves according to the simultaneous gradient:

$$\dot{x}_i = g_i \tag{6.19}$$

$$x_i \leftarrow x_i + \alpha_i \hat{g}_i \tag{6.20}$$

LOLA shapes the anticipated naive updates of opponents based on their individual learning rates η_j:

$$\dot{x}_i = g_i + \sum_{j \neq i} \eta_j \left(\nabla_{x_j} g_i \right) g_j \tag{6.21}$$

$$x_i \leftarrow x_i + \alpha_i \left(\hat{g}_i + \hat{d}_i \left(\sum_{j \neq i} \eta_j z_j^T \hat{g}_j \right) z_i \right) \tag{6.22}$$

CO stabilizes the joint vector field by penalizing it with the transposed Jacobian-vector product scaled by a hyperparameter γ:

$$\dot{x}_i = g_i - \gamma \sum_j \left(\nabla_{x_i} g_j \right) g_j \tag{6.23}$$

$$x_i \leftarrow x_i + \alpha_i \left(\hat{g}_i - \gamma \left(\sum_j \hat{d}_j z_j^T \hat{g}_j \right) z_i \right) \tag{6.24}$$

SGA isolates the rotational dynamics by applying the antisymmetric part of the game Jacobian scaled by λ:

$$\dot{x}_i = g_i + \frac{\lambda}{2} \sum_j \left(\nabla_{x_j} g_i - \nabla_{x_i} g_j \right) g_j \tag{6.25}$$

$$x_i \leftarrow x_i + \alpha_i \left(\hat{g}_i + \frac{\lambda}{2} \left(\hat{d}_i \sum_j z_j^T \hat{g}_j - \sum_j \hat{d}_j z_j^T \hat{g}_j \right) z_i \right) \tag{6.26}$$

By extracting the scalar finite-difference terms upfront, these update rules collapse the cross-agent interactions into simple scalar multipliers applied to the perturbation vector z_i. This results in methods with memory and computational complexity that is strictly linear with respect to the parameter count, while maintaining the constant function evaluation complexity of JPSPG.

6.2 Benchmarks

In this section, we describe our proposed benchmarks. These environments naturally exhibit the complexities outlined in Section 1.2, including infinite heterogeneous populations, matrix-valued

discounting, and discontinuous payoffs. We describe each environment as a POSG, as defined in Section 3.5. Across all benchmarks, we assume I is a non-atomic measure space (for example, the unit interval $[0, 1]$ or unit square $[0, 1]^2$ equipped with the Lebesgue measure), modeling an infinite population of players. Our goal is to show that, for each of these environments, our method converges to Nash equilibrium, as measured by approximate exploitability.

6.2.1 Epidemic: Epidemiological contagion and pandemic mitigation

Drawing from the behavioral epidemiology literature (Reluga, 2010; Acemoglu et al., 2020), this is a mean-field game of viral contagion. Each individual $i \in I$ has a private state $x_i = (v_i, r_i) \in \mathbb{R}^2 = X$ consisting of a viral load v_i and financial reserve r_i. The global state is X^I, consisting of the individuals' private states. The components of the POSG are as follows:

- $I = [0, 1]$ is the non-atomic continuum of individuals.
- $S = X^I$ is the global state space.
- $O_i = X \times \mathbb{R}$ contains the private state x_i alongside a scalar public health signal.
- $A_i = [0, 1]$ is the individual's daily socialization effort.
- $Z : S \to \triangle \Pi O$ is the observation kernel. It provides agent i with $o_i = (x_i, \hat{v})$, where $\hat{v} \sim \mathcal{N}(\bar{v}, \sigma^2)$ is a noisy broadcast of the mean viral load $\bar{v} = \int_{i \sim \mu} v_i$.
- $T : S \times \Pi A \to \triangle S$ is the transition kernel. The viral load v_i evolves stochastically based on a_i and the societal infection density derived from ν. Financial reserves update deterministically as $r_i' = f_i + w_i a_i - c_i$, where w_i and c_i denote cohort-specific wages and living costs, respectively.
- $R : S \times \Pi A \to V$ has a strict dual-discontinuity penalty. The utility R_i suffers a large penalty if the macroscopic mass of severe cases $\int_{i \sim \mu} [\![v_i > v_{\text{crit}}]\!]$ strictly exceeds a healthcare capacity C, or if the agent faces bankruptcy ($r_i \leq 0$).
- $\Gamma : S \times \Pi A \times S \to \mathcal{L}(V)$ is an integral operator defining demographic altruism, with a kernel $K(i, j)$ that specifies how $i \in I$ discounts the future returns of $j \in I$.
- $\rho \in \triangle S$ initializes the global state to reflect a localized outbreak.

6.2.2 Finance: Financial contagion and systemic risk

Inspired by models of systemic risk (Eisenberg and Noe, 2001; Elliott, Golub, and Jackson, 2014), this game models a global financial network. Let $X = \mathbb{R}^k$ represent the private balance sheet of institution i, capturing $k \in \mathbb{N}$ distinct balance sheet variables, including capital reserves c_i and toxic asset holdings w_i. To capture exogenous market forces, let $M = \mathbb{R}^m$ represent $m \in \mathbb{N}$ global macroeconomic indices. The components of the POSG are as follows:

- $I = [0, 1]$ is the continuum of financial institutions.
- $S = X^I \times M$ couples the microscopic balance sheets with the independent macroeconomic indicators.
- $O_i = X \times M$ reflects the fact that an institution observes its own balance sheet (one copy of X) and the global indices M, but not the balance sheets of other institutions.
- $A_i \in \mathbb{R}^d$ denotes the continuous capital allocation vectors for interbank lending and liquidation.

- $Z : S \to \triangle \Pi O$ exposes private state components of i to i.
- $T : S \times \Pi A \to \triangle S$ is a **jump-diffusion process** updating both balance sheets and global indices. Crucially, capital defaults trigger discontinuous cascading shocks across neighboring transition probabilities.
- $R : S \times \Pi A \to V$ imposes regulatory discontinuities. If capital reserves c_i fall below a statutory threshold θ, R_i evaluates strictly to zero and a forced liquidation is triggered.
- $\Gamma : S \times \Pi A \times S \to \mathcal{L}(V)$ acts as a linear operator that dynamically redistributes discounted returns in proportion to institutional equity ownership percentages.
- $\rho \in \triangle S$ initializes the market with a baseline distribution of interconnected assets and capital.

6.2.3 Grid: Wholesale forward capacity markets

Drawing from mean-field models of smart grids (Borenstein, Bushnell, and Knittel, 1999; Kizilkale, Salhab, and Malhamé, 2019), this game models a dynamic electricity market. Let $X = \mathbb{R}_{\geq 0}$ represent the physically constrained megawatt capacity c_i of generator i. The components of the POSG are as follows:

- $I = [0, 1]$ is the population of heterogeneous power generators.
- $S = X^I \times \mathbb{R}_{\geq 0}$ couples the microscopic capacities with a global grid frequency.
- $O_i = X \times \mathbb{R}_{\geq 0}$ provides local capacity measurements alongside a stochastically noisy reading of grid frequency.
- $A_i \in \mathbb{R}^2_{\geq 0}$ represents the continuous parameters of a step-function supply bid, comprising an offered volume $v_i \leq c_i$ and a minimum acceptable price p_i. These yield the individual supply curve $S_i(P) = v_i[\![P \geq p_i]\!]$.
- $Z : S \to \triangle \Pi O$ obfuscates the global supply curve and competing bids.
- $T : S \times \Pi A \to \triangle S$ evolves individual capacities c_i via mean-reverting **Ornstein-Uhlenbeck processes** representing weather stochasticity, while the grid frequency updates reactively based on aggregate market bid volumes.
- $R : S \times \mathbf{A} \to V$ imposes a strict market-clearing discontinuity via a uniform-price auction. Let $D \in \mathbb{R}_{\geq 0}$ represent the macroscopic inelastic electricity demand. The environment computes the aggregate supply curve by integrating the step-function bids across the continuum: $S(P) = \int_{j \sim I} v_j \mathbb{I}(P \geq p_j)$. The market establishes the uniform clearing price P^* at the exact intersection of supply and demand:

$$P^* = \inf\{P \in \mathbb{R}_{\geq 0} : S(P) \geq D\} \tag{6.27}$$

  The reward R_i for generator i is then evaluated as a piecewise function, creating a sharp discontinuity for bids exceeding the clearing price:

$$R_i(s, \mathbf{a}) = \begin{cases} v_i P^* & \text{if } p_i \leq P^* \\ 0 & \text{if } p_i > P^* \end{cases} \tag{6.28}$$

- $\Gamma : S \times \Pi A \times S \to \mathcal{L}(V)$ encodes regional stability pacts, internalizing the grid stability of physically interconnected neighbors.
- $\rho \in \triangle S$ samples the baseline weather states and initial generator capacities.

6.2.4 Foraging: Evolutionary foraging with kin selection

Drawing from the foundations of evolutionary game theory (Hamilton, 1964; McNamara and Houston, 1996), we formulate a spatial foraging game. Let $X = \mathbb{R}_{\geq 0} \times \mathbb{R}^2$ define the private biological state, comprising internal energy reserves e_i and 2D spatial coordinates p_i. Let $E = L^2(\mathbb{R}^2, \mathbb{R}_{\geq 0})$ be the independent, continuous spatial density field of regenerating nutrients. The components of the POSG are as follows:

- $I = [0, 1]$ is the continuum of competing biological organisms.
- $S = X^I \times E$ combines the microscopic states of organisms with the global nutrient field.
- $O_i = \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$ equips the agent with knowledge of its internal energy e_i and a localized sensory integral of nearby nutrients centered at p_i.
- $A_i \in \mathbb{R}^2$ is the continuous velocity vector dictating spatial movement.
- $Z : S \to \triangle\Pi O$ restricts spatial observation to an area around p_i.
- $T : S \times \Pi A \to \triangle S$ updates spatial coordinates $p_i' = p_i + a_i$, depletes energy $e_i' = e_i - c\|a_i\|$, and stochastically regenerates the encompassing nutrient field E.
- $R : S \times \Pi A \to V$ enforces a biological starvation discontinuity. Utility scales with successful nutrient intake, but becomes permanently zero if the internal energy e_i falls below a critical survival threshold e_{crit}.
- $\Gamma : S \times \Pi A \times S \to \mathcal{L}(V)$ weighs the returns of agent j in agent i's utility by genetic relatedness r_{ij}, realizing **Hamilton's rule** of kin selection[1], weighing the returns of genetically-similar agents.
- $\rho \in \triangle S$ samples the initial spatial distribution of both agents and available nutrients.

6.2.5 Orbit: Orbital constellation management and debris avoidance

Inspired by the physical challenges of low Earth orbit mega-constellations (Kessler and Cour-Palais, 1978; Alfriend et al., 2009), we model an orbital routing game. Let $X = \mathbb{R}^6 \times \mathbb{R}_{\geq 0}$ represent the private state, capturing the 6D orbital vector q_i (position and velocity) alongside finite chemical fuel reserves f_i. The components of the POSG are as follows:

- $I = [0, 1]$ represents the continuum of active satellites interspersed with untrackable micro-debris.
- $S = X^I$ distills the global state entirely to the microscopic profile of all orbital objects.
- $O_i = X \times \mathbb{R}_{\geq 0}$ provides the exact internal state and noisy radar estimations of proximate objects.
- $A_i \in \mathbb{R}^3$ dictates the continuous thrust vector allocated for station-keeping maneuvers.
- $Z : S \to \triangle\Pi O$ simulates radar noise, sensor degradation, and periodic line-of-sight signal occlusion.

[1] Hamilton's rule of kin selection (Hamilton, 1964) is a mathematical formula that explains why an animal might act altruistically, sacrificing its own fitness to help another, even when it seems to go against its own survival. It suggests that "altruistic" genes can evolve if the benefit to a relative is high enough to outweigh the cost to the individual. The rule is that genes for a particular behavior should increase in frequency when $rB > C$, where r is the genetic relatedness of the recipient to the actor, B is the additional reproductive benefit gained by the recipient of the altruistic act, and C is the reproductive cost to the individual performing the act.

- $T : S \times \Pi A \to \triangle S$ integrates non-linear orbital mechanics to constantly update q_i, while strictly depleting fuel as $f_i' = f_i - c\|a_i\|$, where c represents the specific fuel consumption (or the inverse of the propulsion system's efficiency).
- $R : S \times \Pi A \to V$ models the Kessler syndrome[2]. The utility R_i evaluates to a massive penalty if the spatial distance between q_i and any q_j drops below a physical collision radius, or if the satellite burns through all remaining fuel ($f_i \leq 0$).
- $\Gamma : S \times \Pi A \times S \to \mathcal{L}(V)$ represents overarching geopolitical alliances, assigning high positive weights to allied orbital assets and non-positive weights to adversaries.
- $\rho \in \triangle S$ initializes orbits derived directly from empirical aerospace tracking catalogs.

6.2.6 Traffic: Macroscopic urban traffic and fleet routing

Drawing from mean-field routing games (Wardrop, 1952; Bressan, 2015), this game models the fluid dynamics of continuous urban traffic. Let $X = \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}_{\geq 0}$ represent the state of driver i, structurally containing position p_i, velocity v_i, and an elapsed trip time t_i. The empirical spatial density field ν is derived from the profile of all driver positions. The components of the POSG are as follows:

- $I = [0, 1]$ is the continuum of vehicles navigating the road network.
- $S = X^I$ maintains the global state strictly through the kinematics and trip times of all individual agents.
- $O_i = X \times \mathbb{R}_{\geq 0}$ is an observation of local kinematics alongside a noisy local measurement of ν.
- $A_i \in \mathbb{R}^2$ represents the combined acceleration and steering vector.
- $Z : S \to \triangle \Pi O$ limits observation to the local density.
- $T : S \times \Pi A \to \triangle S$ updates p_i and v_i subject to a non-linear drag generated directly by the local density derived from ν. Elapsed time t_i increments deterministically.
- $R : S \times \Pi A \to V$ implements gridlock and deadline discontinuities. The utility R_i monotonically decays with trip time t_i, but drops if local density breaches a maximum flow threshold or if t_i violates a strict delivery deadline.
- $\Gamma : S \times \Pi A \times S \to \mathcal{L}(V)$ enables ride-sharing fleet coordination, distributing discounted returns across vehicles operating within the same corporate network, which have an incentive to optimize *global* systemic throughput.
- $\rho \in \triangle S$ initializes the macroscopic spatial profile of vehicles by sampling from historically-grounded spatial gravity and bottleneck models (Voorhees, 1955; Wilson, 1967; Vickrey, 1969; Arnott, De Palma, and Lindsey, 1990), generating realistic, asymmetrical origin-destination demand across the density field.

[2]Kessler syndrome (Kessler and Cour-Palais, 1978) describes a situation in which the density of objects in low Earth orbit (LEO) becomes so high due to space pollution that collisions between these objects cascade, exponentially increasing the amount of space debris over time.

6.3 Codebase

Our codebase is written in **Python** (Rossum and Drake, 1995) and uses the following libraries:

- **JAX** (Bradbury et al., 2018): a library for high-performance numerical computing and large-scale machine learning.
- **Flax** (Heek et al., 2023): a neural network library and ecosystem for JAX designed for flexibility.
- **Optax** (DeepMind et al., 2020): a gradient processing and optimization library for JAX. It contains implementations of the most common optimizers used in machine learning.
- **Diffrax** (Kidger, 2021): a JAX-based library providing numerical differential equation solvers that is autodifferentiable and GPU-capable.
- **Mctx** (DeepMind et al., 2020): JAX-native implementation of Monte Carlo tree search (MCTS) algorithms.
- **Matplotlib** (Hunter, 2007): a comprehensive library for creating static, animated, and interactive visualizations.
- **SciPy** (Virtanen et al., 2020): an open-source software for mathematics, science, and engineering. It includes modules for statistics, optimization, integration, linear algebra, Fourier transforms, signal and image processing, ODE solvers, and more.
- **NumPy** (Harris et al., 2020): the fundamental package for scientific computing with Python.
- **Pandas** (The Pandas development team, 2020; McKinney, 2010): a flexible and powerful data analysis / manipulation library for Python, providing labeled data structures.
- **PyCairo**: Python bindings for the Cairo graphics library (Worth and Packard, 2003).
- **CVXPY** (Diamond and Boyd, 2016; Agrawal et al., 2018): a Python-embedded modeling language for convex optimization problems.
- **OTT-JAX** (Cuturi et al., 2022): JAX package providing tools for solving optimal transport problems (e.g., Wasserstein distance between point clouds).
- **h5py**: a thin, Pythonic wrapper around HDF5 (Koranne, 2010; The HDF Group, 2024), a high-performance data management and storage suite.

I have extensive experience with Python, JAX, and all of these libraries. I use them daily to code my environments, agents, and learning algorithms. For evidence of this, below are my open-source contributions to JAX and Optax:

- `https://github.com/jax-ml/jax/pulls?q=is:pr+author:carlosgmartin`
- `https://github.com/google-deepmind/optax/pulls?q=is:pr+author:carlosgmartin`

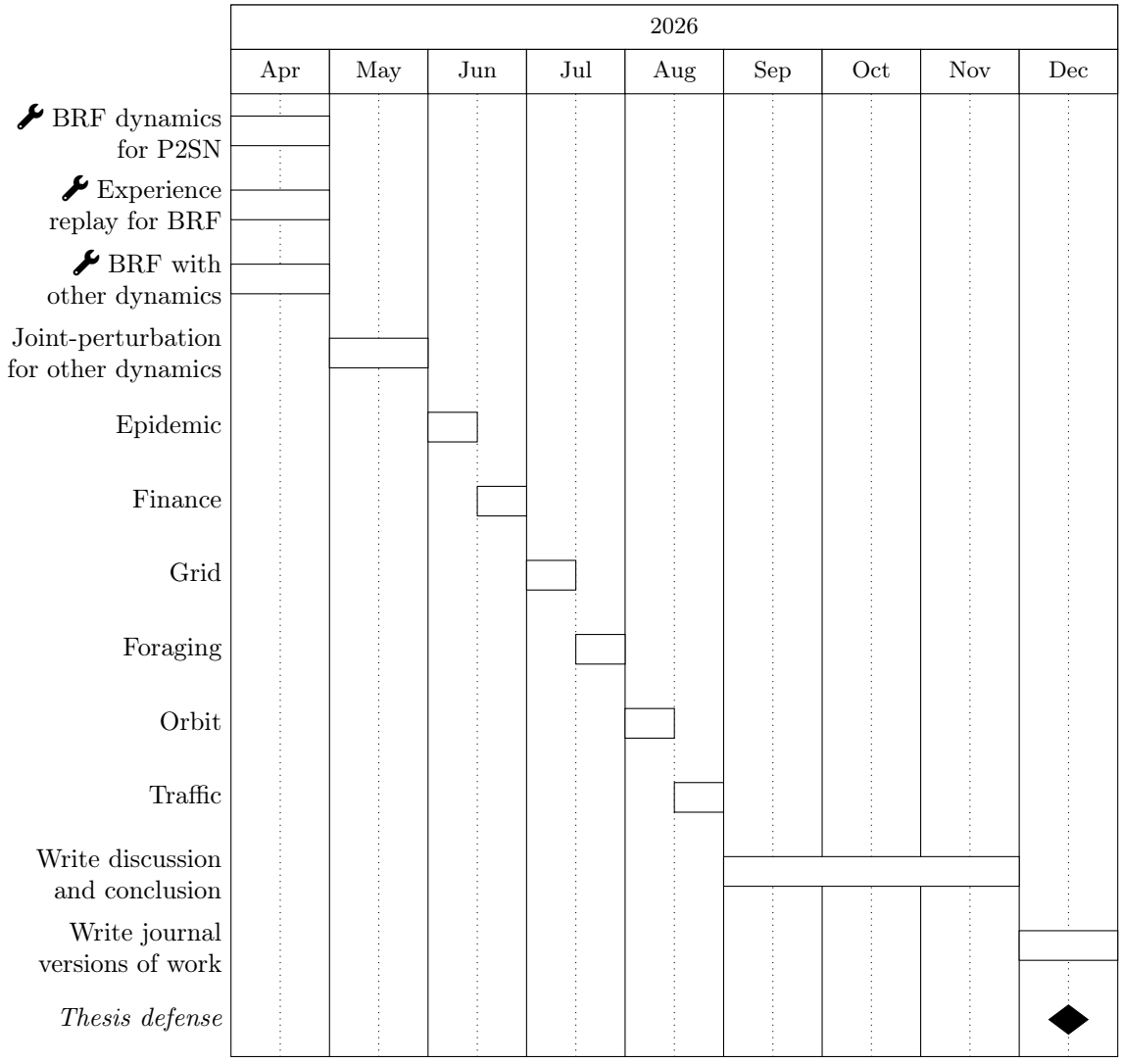
6.4 Available compute

My lab has a dedicated computing cluster, the Marvel computing cluster. It has 1 node with an AMD EPYC 7742 64-Core Processor and 8 NVIDIA A100 GPUs. It also has 14 other nodes, each with an AMD EPYC 7282 16-Core Processor. It also has 5 other nodes, each with an AMD Opteron Processor 6272.

I also have access to the Google Cloud ORCHARD Cluster managed by CMU. Each job can run for 12 hours and has access to a maximum of 1 GPU.

6.5 Timeline

My proposed timeline is as follows. As stated before, most of the code and writeup is **already done**. Also, experiments are running continuously in the background on my lab's cluster, with occasional monitoring, so experiments can overlap with either the programming or the writeup. The symbol 🔧 is used to mark work that is almost complete.



6.6 Stretch goals

The contributions outlined above form the foundational core of this thesis and are sufficient to validate it. However, if time and computational resources permit, I also plan to explore the following optional extensions. While not strictly necessary for the completion of the dissertation, these speculative directions build naturally upon my core framework, and represent potentially interesting avenues for future research.

- Extend to **alternative solution concepts**. These include the following:
  - Correlated equilibria (CE) (Aumann, 1974).
  - Coarse correlated equilibria (CCE) (Moulin and Vial, 1978).
  - Quantal response equilibria (QRE) (McKelvey and Palfrey, 1995; McKelvey and Palfrey, 1998).
  - α-rank (Omidshafiei et al., 2019).
- Extend to **mechanism design**, that is, tuning game parameters so that the resulting equilibria have desirable properties from the perspective of a social planner. There is a vast literature on mechanism design. For this, we could use **bilevel optimization** techniques. The state of the art in continuous bilevel optimization uses implicit differentiation and variance-reduced stochastic approximation to efficiently estimate inverse Hessians and compute hypergradients at scale (Lorraine, Vicol, and Duvenaud, 2020; Ji, Yang, and Liang, 2021). For discrete or mixed-integer settings, state-of-the-art methods use specialized branch-and-bound frameworks applied to Karush–Kuhn–Tucker (KKT) or value-function reformulations (Kleinert et al., 2021).
- Extend to **inverse multiagent reinforcement learning** (IRL). This means inferring characteristics of a game from players' behavior. I have already done some work along these lines in Martin and Sandholm (2021a). To scale this up to more general games where exact Bayesian inference is intractable, we could use some tractable approximations. For example, we could use the following techniques.
  - Markov chain Monte Carlo (MCMC) (Metropolis et al., 1953; Hastings, 1970) is a class of algorithms that enables sampling from complex, high-dimensional probability distributions by constructing a Markov chain that has the desired distribution as its equilibrium state. These include the Metropolis–Hastings (MH) algorithm (Hastings, 1970; Metropolis et al., 1953), Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal, 1996), and No-U-Turn Sampling (NUTS) (Hoffman and Gelman, 2014).
  - Dropout (Srivastava et al., 2014) is an effective way to represent model uncertainty in deep learning (Gal and Ghahramani, 2016).
  - Likewise, DropConnect (Wan et al., 2013) is effective in modeling uncertainty of Bayesian deep networks (Mobiny et al., 2021).
- Extend to **infinite utilities**. To be more precise, these would be non-Archimedean utilities, under lexicographic ordering or leximin ordering. These could be handled via **multilevel optimization**.
- Incorporate **efficient exploration** techniques, as we did in Martin and Sandholm (2021b). There is a vast literature on efficient exploration for RL and MARL. We could try to generalize it and apply it to our infinite setting.

- Incorporate **execution-time planning** into the agents, as we did in Martin, Boutilier, et al. (2024) and Martin and Sandholm (2025a).

In addition, we include the following environment as a stretch goal.

6.6.1 Auction: Real-time programmatic ad auctions

We adapt continuous real-time bidding (Edelman, Ostrovsky, and Schwarz, 2007; Cai, Ren, et al., 2017) into a multi-step POSG. Let $X = \mathbb{R}^2_{\geq 0}$ represent the private state of advertiser i, consisting of a depleting remaining budget b_i and a hidden conversion valuation v_i. The components of the POSG are as follows:

- $I = [0, 1]$ is the continuum of autonomous advertising agents.
- $S = X^I$ is the global state consisting of the agents' profile of budgets and valuations.
- $O_i = \mathbb{R}_{\geq 0}$ restricts observation, providing the agent only with its current budget b_i while hiding its true conversion rate and the states of competitors.
- $A_i \in \mathbb{R}_{\geq 0}$ is the continuous financial bid submitted for an impression, strictly bounded by the remaining budget ($a_i \leq b_i$).
- $Z : S \to \triangle \Pi O$ acts as an information filter, obscuring the latent macroscopic demographic variables D_t that govern the stochastic evolution of the true conversion valuation v_i, forcing the agent to infer market regimes strictly from delayed auction outcomes.
- $T : S \times \Pi A \to \triangle S$ deterministically subtracts the clearing price from the winning agent's budget b_i and stochastically evolves latent valuations v_i via a hidden Markov model.
- $R : S \times \Pi A \to V$ models a generalized second-price auction mechanism. The utility yields $R_i = v_i - \max_{j \neq i} a_j$ if $a_i > \max_{j \neq i} a_j$, and zero otherwise.
- $\Gamma : S \times \Pi A \times S \to \mathcal{L}(V)$ models advertising agency portfolios, coupling the returns of client brands operating under the same centralized bidding algorithm.
- $\rho \in \triangle S$ assigns the initial campaign budgets and baseline valuation distributions.

In programmatic advertising, the true value of an ad impression (the conversion rate) is not static. It is driven by macroscopic, shifting demographic trends, such as a viral social media trend making teenagers suddenly buy a specific product, a seasonal purchasing behavior, or a competitor launching a rival ad campaign. Advertisers do not get to see these underlying demographic shifts directly. They only see the downstream effects: whether their bid won, and whether that win eventually resulted in a conversion. The market is constantly shifting beneath their feet, and they must infer those shifts purely from incomplete, noisy feedback. To formalize this, we can model the demographic shifts as a latent stochastic process. Let $D_t \in \mathbb{R}^m$ represent a hidden macroscopic vector tracking $m \in \mathbb{N}$ demographic/market regimes at time t. The true, hidden valuation $v_{i,t}$ is no longer a static number, but a dynamic function of this latent demographic state: $v_{i,t} = f_i(D_t) + \varepsilon$, where ε is a noise term.

Bibliography

Acemoglu, Daron, Victor Chernozhukov, Iván Werning, and Michael D. Whinston (2020). "A multi-risk SIR model with optimally targeted lockdown". In: *NBER Working Paper Series*.

Adam, Lukáš, Rostislav Horčík, Tomáš Kasl, and Tomáš Kroupa (2021). "Double oracle algorithm for computing equilibria in continuous games". In: *AAAI Conference on Artificial Intelligence*.

Adamo, Tim and Alexander Matros (2009). "A Blotto game with incomplete information". In: *Economics Letters*.

Aggarwal, Gagan and Jason D. Hartline (2006). "Knapsack auctions". In: *SODA*.

Agrawal, Akshay, Robin Verschueren, Steven Diamond, and Stephen Boyd (2018). "A rewriting system for convex optimization problems". In: *Journal of Control and Decision*.

Alfriend, Kyle T., Srinivas R. Vadali, Pini Gurfil, Jonathan P. How, and Louis S. Breger (2009). *Spacecraft formation flying: dynamics, control and navigation*. Elsevier.

Arnott, Richard, André De Palma, and Robin Lindsey (1990). "Economics of a bottleneck". In: *Journal of urban economics*.

Aumann, Robert (1964). "Markets with a continuum of traders". In: *Econometrica*.

— (1974). "Subjectivity and correlation in randomized strategies". In: *Journal of Mathematical Economics*.

Bachmann, Paul (1894). *Die Analytische Zahlentheorie (Analytic Number Theory)*. Teubner.

Bäck, Thomas (1996). *Evolutionary algorithms in theory and practice*. Oxford University Press.

Bäck, Thomas, David B. Fogel, and Zbigniew Michalewicz (1997). *Handbook of evolutionary computation*. IOP Publishing Ltd.

Bae, Juhan and Roger B. Grosse (2020). "Delta-STN: efficient bilevel optimization for neural networks using structured response Jacobians". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Bailey, Bolton and Matus Telgarsky (2018). "Size-noise tradeoffs in generative networks". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Balduzzi, David, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel (2018). "The mechanics of n-player differentiable games". In: *International Conference on Machine Learning (ICML)*.

Bao, Xuchan and Guodong Zhang (2022). "Finding and only finding local Nash equilibria by both pretending to be a follower". In: *Workshop on Gamification and Multiagent Solutions*.

Bardi, Martino and Italo Capuzzo-Dolcetta (1997). *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman Equations*. Birkhäuser Boston.

Başar, Tamer and Geert Jan Olsder (1999). *Dynamic noncooperative game theory*. SIAM.

Baye, Michael R., Dan Kovenock, and Casper G. de Vries (1996). "The all-pay auction with complete information". In: *Economic Theory*.

Berahas, Albert, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg (2022). "A theoretical and empirical comparison of gradient approximations in derivative-free optimization". In: *Foundations of Computational Mathematics*.

Berg, Jordan, Amy Greenwald, Victor Naroditskiy, and Eric Sodomka (2010). "A knapsack-based approach to bidding in ad auctions". In: *European Conference on Artificial Intelligence (ECAI)*. IOS Press.

Berger, Ulrich (2007). "Brown's original fictitious play". In: *Journal of Economic Theory (JET)*.

Bergman, L. M. and I. N. Fokin (1998). "On separable non-cooperative zero-sum games". In: *Optimization*.

Berner, Christopher, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Jozefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d.O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang (2019). "Dota 2 with large scale deep reinforcement learning". In: *arXiv:1912.06680*.

Berridge, Steffan and Jacek B. Krawczyk (1997). *Relaxation algorithms in finding Nash equilibria*. Tech. rep. Victoria University of Wellington.

Berry, Andrew C. (1941). "The accuracy of the Gaussian approximation to the sum of independent variates". In: *Transactions of the American Mathematical Society*.

Bewley, Truman F. (1972). "Existence of equilibria in economies with infinitely many commodities". In: *Journal of Economic Theory (JET)*.

Bichler, Martin, Max Fichtl, and Matthias Oberlechner (2023). "Computing Bayes–Nash equilibrium strategies in auction games via simultaneous online dual averaging". In: *Operations Research*.

Bichler, Martin, Maximilian Fichtl, Stefan Heidekrüger, Nils Kohring, and Paul Sutterer (2021). "Learning equilibria in symmetric auction games using artificial neural networks". In: *Nature Machine Intelligence*.

Bichler, Martin, Nils Kohring, and Stefan Heidekrüger (2023). "Learning equilibria in asymmetric auction games". In: *INFORMS Journal on Computing*.

Bochner, Salomon (1933). "Integration von Funktionen, deren Werte die Elemente eines Vektorraumes sind". In: *Fundamenta Mathematicae*.

Boix-Adserà, Enric, Benjamin L. Edelman, and Siddhartha Jayanti (2021). "The multiplayer Colonel Blotto game". In: *Games and Economic Behavior (GEB)*.

Bolton, Gary E. and Axel Ockenfels (2000). "ERC: A theory of equity, reciprocity, and competition". In: *American Economic Review (AER)*.

Bonabeau, Eric (2002). "Agent-based modeling: methods and techniques for simulating human systems". In: *Proceedings of the National Academy of Sciences (PNAS)*.

Borel, Émile (1921). "The theory of play and integral equations with skew symmetric kernels". In: *Comptes Rendus de l'Académie des Sciences*.

Borel, Émile and Jean Ville (1938). *Traité du calcul des probabilités et ses applications*. Gauthier-Villars.

Borenstein, Severin, James Bushnell, and Christopher R. Knittel (1999). "Market power in electricity markets: beyond concentration measures". In: *The Energy Journal*.

Bradbury, James, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang (2018). *JAX: composable transformations of Python+NumPy programs*.

Brandt, Felix, Tuomas Sandholm, and Yoav Shoham (2007). "Spiteful bidding in sealed-bid auctions". In: *International Joint Conference on Artificial Intelligence (IJCAI)*.

Bravo, Mario, David Leslie, and Panayotis Mertikopoulos (2018). "Bandit learning in concave N-person games". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Bressan, Alberto (2015). "Conservation law models for traffic flow on a network of roads". In: *Networks and Heterogeneous Media*.

Brouwer, Luitzen Egbertus Jan (1911). "Über abbildung von mannigfaltigkeiten". In: *Mathematische annalen*.

Brown, George W. (1951). "Iterative solution of games by fictitious play". In: *Activity Analysis of Production and Allocation*. Wiley.

Brown, Noam and Tuomas Sandholm (2018). "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals". In: *Science*.

— (2019). "Superhuman AI for multiplayer poker". In: *Science*.

Bunyakovsky, Viktor Y. (1859). "Sur quelques inégalités concernant les intégrales aux différences finis". In: *Mémoires de l'Académie Impériale des Sciences de St. Pétersbourg*.

Busoniu, Lucian, Robert Babuska, and Bart De Schutter (2008). "A comprehensive survey of multiagent reinforcement learning". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*.

Cai, Han, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo (2017). "Real-time bidding by reinforcement learning in display advertising". In: *ACM International Conference on Web Search and Data Mining (WSDM)*.

Cai, Yang, Ozan Candogan, Constantinos Daskalakis, and Christos Papadimitriou (2016). "Zero-sum polymatrix games: a generalization of minmax". In: *Mathematics of Operations Research*.

Cai, Yang and Constantinos Daskalakis (2011). "On minmax theorems for multiplayer games". In: *ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

Caines, Peter and Minyi Huang (2021). "Graphon mean field games and their equations". In: *SIAM Journal on Control and Optimization*.

Caines, Peter, Minyi Huang, and Roland Malhamé (2018). "Mean field games". In: *Handbook of Dynamic Game Theory*. Springer International Publishing.

Cauchy, Augustin Louis Baron (1821a). *Cours d'analyse de l'École Royale Polytechnique*. Imprimerie royale.

Cauchy, Augustin-Louis (1821b). "Sur les formules qui résultent de l'emploie du signe et sur$>$ ou$<$, et sur les moyennes entre plusieurs quantités". In: *Cours d'Analyse, 1er Partie: Analyse algébrique*.

Chan, Patrick and Ronnie Sircar (2015). "Bertrand and Cournot mean field games". In: *Applied Mathematics & Optimization*.

Charness, Gary and Matthew Rabin (2002). "Understanding social preferences with simple tests". In: *The Quarterly Journal of Economics*.

Chen, Bill and Jerrod Ankenman (2006). *The Mathematics of Poker*. ConJelCo.

Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter (2016). "Fast and accurate deep network learning by exponential linear units (ELUs)". In: *International Conference on Learning Representations (ICLR)*.

Cloud, Alex, Albert Wang, and Wesley Kerr (2023). "Anticipatory fictitious play". In: *International Joint Conference on Artificial Intelligence (IJCAI)*.

Cornes, Richard and Roger Hartley (2007). "Aggregative public good games". In: *Journal of Public Economic Theory*.

Coulom, Rémi (2007). "Efficient selectivity and backup operators in Monte–Carlo tree search". In: *Computers and Games*. Springer.
Cournot, Antoine Augustin (1838). *Recherches sur les principes mathématiques de la théorie des richesses*. Hachette.
— (1863). *Principes de la théorie des richesses*. Hachette.
Crouse, David F. (2016). "On implementing 2D rectangular assignment algorithms". In: *IEEE Transactions on Aerospace and Electronic Systems*.
Cui, Kai and Heinz Koeppl (2022). "Learning graphon mean field games and approximate Nash equilibria". In: *International Conference on Learning Representations (ICLR)*.
Cuturi, Marco, Laetitia Meng-Papaxanthos, Yingtao Tian, Charlotte Bunne, Geoff Davis, and Olivier Teboul (2022). "Optimal transport tools (OTT): a JAX Toolbox for all things Wasserstein". In: *arXiv:2201.12324*.
Cybenko, George (1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems*.
Dasgupta, Partha and Eric Maskin (1986). "The existence of equilibrium in discontinuous economic games 1: theory". In: *Review of Economic Studies*.
Daskalakis, Constantinos, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng (2018). "Training GANs with optimism". In: *International Conference on Learning Representations (ICLR)*.
DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Miloš Stanojević, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola (2020). *The DeepMind JAX ecosystem*.
Deng, Li (2012). "The MNIST database of handwritten digit images for machine learning research". In: *IEEE Signal Processing Magazine*.
Diamond, Steven and Stephen Boyd (2016). "CVXPY: a Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research*.
Dockner, Engelbert, Steffen Jorgensen, Van Long Ngo, and Gerhard Sorger (2000). *Differential games in economics and management science*. Cambridge University Press.
Domingo-Enrich, Carles (2019). "Games in machine learning: differentiable n-player games and structured planning". MA thesis. Universitat Politècnica de Catalunya.
Dormand, John R. and Peter J. Prince (1980). "A family of embedded Runge–Kutta formulae". In: *Journal of computational and applied mathematics*.
Dror, Moshe (1989). "Simple proof for Goofspiel: the game of pure strategy". In: *Advances in Applied Probability*.
Duane, Simon, Anthony D. Kennedy, Brian J. Pendleton, and Duncan Roweth (1987). "Hybrid Monte Carlo". In: *Physics letters B*.
Duchi, John C., Peter L. Bartlett, and Martin J. Wainwright (2012). "Randomized smoothing for stochastic optimization". In: *SIAM Journal on Optimization*.
Duchi, John C., Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono (2015). "Optimal rates for zero-order convex optimization: the power of two function evaluations". In: *IEEE Transactions on Information Theory*.

Dunford, Nelson (1935). "Integration in general analysis". In: *Transactions of the American Mathematical Society*.
Dütting, Paul, Vasilis Gkatzelis, and Tim Roughgarden (2014). "The performance of deferred-acceptance auctions". In: *ACM Conference on Economics and Computation (EC)*.
Edelman, Benjamin, Michael Ostrovsky, and Michael Schwarz (2007). "Internet advertising and the generalized second-price auction: selling billions of dollars worth of keywords". In: *American Economic Review (AER)*.
Efron, Bradley (1979). "Bootstrap methods: another look at the jackknife". In: *The Annals of Statistics*.
— (1987). "Better bootstrap confidence intervals". In: *Journal of the American Statistical Association (JASA)*.
Eiben, Agoston E. and James E. Smith (2003). *Introduction to evolutionary computing*. Springer.
Eisenberg, Larry and Thomas H. Noe (2001). "Systemic risk in financial systems". In: *Management Science*.
Elliott, Matthew, Benjamin Golub, and Matthew O. Jackson (2014). "Financial networks and contagion". In: *American Economic Review (AER)*.
Epstein, Joshua and Robert Axtell (1996). *Growing artificial societies: social science from the bottom up*. Brookings Institution Press.
Fehr, Ernst and Klaus M. Schmidt (1999). "A theory of fairness, competition, and cooperation". In: *The Quarterly Journal of Economics*.
Feldman, William, Inwon Kim, and Aaron Zeff Palmer (2024). "The sharp interface limit of an Ising game". In: *ESAIM: Control, Optimisation and Calculus of Variations*.
Feng, Ruili, Deli Zhao, and Zheng-Jun Zha (2021). "Understanding noise injection in GANs". In: *International Conference on Machine Learning (ICML)*.
Ferguson, Thomas S. and Costis Melolidakis (2001). "Games with finite resources". In: *International Journal of Game Theory (IJGT)*.
Fiez, Tanner, Benjamin Chasnov, and Lillian Ratliff (2020). "Implicit learning dynamics in stackelberg games: equilibria characterization, convergence analysis, and empirical study". In: *International Conference on Machine Learning (ICML)*.
Fiez, Tanner, Benjamin Chasnov, and Lillian J. Ratliff (2019). "Convergence of learning dynamics in Stackelberg games". In: *arXiv:1906.01217*.
Fiez, Tanner, Chi Jin, Praneeth Netrapalli, and Lillian J. Ratliff (2022). "Minimax optimization with smooth algorithmic adversaries". In: *International Conference on Learning Representations (ICLR)*.
Flåm, Sjur Didrik and Anatoly S. Antipin (1996). "Equilibrium programming using proximal-like algorithms". In: *Mathematical Programming*.
Flåm, Sjur Didrik and Andrzej Ruszczyński (2008). "Finding normalized equilibrium in convex-concave games". In: *International Game Theory Review*.
Foerster, Jakob N., Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch (2018). "Learning with opponent-learning awareness". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
French, Robert M. (1999). "Catastrophic forgetting in connectionist networks". In: *Trends in cognitive sciences*.
Friedman, Avner (1971a). "Computation of saddle points for differential games of pursuit and evasion". In: *Archive for Rational Mechanics and Analysis*.
— (1971b). *Differential games*. Wiley-Interscience.

Fu, Michael C., ed. (2015). *Handbook of simulation optimization*. Springer.
Fudenberg, Drew and Jean Tirole (1991). *Game theory*. MIT Press.
Gadetsky, Artyom, Kirill Struminsky, Christopher Robinson, Novi Quadrianto, and Dmitry Vetrov (2020). "Low-variance black-box gradient estimates for the Plackett–Luce distribution". In: *AAAI Conference on Artificial Intelligence*.
Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a Bayesian approximation: representing model uncertainty in deep learning". In: *International Conference on Machine Learning (ICML)*.
Galam, Serge and Bernard Walliser (2010). "Ising model versus normal form game". In: *Physica A*.
Ganzfried, Sam (2021). "Algorithm for computing approximate Nash equilibrium in continuous games with application to continuous Blotto". In: *Games*.
Ganzfried, Sam and Tuomas Sandholm (2010). "Computing equilibria by incorporating qualitative models". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
Gemp, Ian and Sridhar Mahadevan (2018). "Global convergence to the equilibrium of GANs using variational inequalities". In: *arXiv:1808.01531*.
Gemp, Ian, 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: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
Ghosh, Papiya and Rajendra P. Kundu (2019). "Best-shot network games with continuous action space". In: *Research in Economics*.
Gilles, Christian and Stephen F. LeRoy (1992). "Bubbles and charges". In: *International Economic Review*.
Glicksberg, Irving Leonard (1952). "A further generalization of the Kakutani fixed point theorem, with application to Nash equilibrium points". In: *Proceedings of the American Mathematical Society*.
Glicksberg, Irving Leonard and Oliver Alfred Gross (1953). "Notes on games over the square". In: *Contributions to the Theory of Games*.
Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*.
Goktas, Denizalp and Amy Greenwald (2021). "Convex-concave min-max Stackelberg games". In: *Conference on Neural Information Processing Systems (NeurIPS)*.
— (2022a). "Exploitability minimization in games and beyond". In: *Conference on Neural Information Processing Systems (NeurIPS)*.
— (2022b). "Gradient descent ascent in min-max Stackelberg games". In: *arXiv:2208.09690*.
Goktas, Denizalp, Jiayi Zhao, and Amy Greenwald (2022a). "Robust no-regret learning in min-max Stackelberg games". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
Goktas, Denizalp, Sadie Zhao, and Amy Greenwald (2022b). "Zero-sum stochastic Stackelberg games". In: *Conference on Neural Information Processing Systems (NeurIPS)*.
Goldblum, Micah, Marc Finzi, Keefer Rowan, and Andrew Gordon Wilson (2024). "Position: the no free lunch theorem, Kolmogorov complexity, and the role of inductive biases in machine learning". In: *International Conference on Machine Learning (ICML)*.
Golowich, Noah, Sarath Pattathil, Constantinos Daskalakis, and Asuman Ozdaglar (2020). "Last iterate is slower than averaged iterate in smooth convex-concave saddle point problems". In: *Conference on Learning Theory (COLT)*.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative adversarial nets". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Grimes, Mark and Moshe Dror (2013). "Observations on strategies for Goofspiel". In: *IEEE CIG*.

Grimm, Volker, Eloy Revilla, Uta Berger, Florian Jeltsch, Wolf M. Mooij, Steven F. Railsback, Hans-Hermann Thulke, Jacob Weiner, Thorsten Wiegand, and Donald L. DeAngelis (2005). "Pattern-oriented modeling of agent-based complex systems: lessons from ecology". In: *Science*.

Grnarova, Paulina, Yannic Kilcher, Kfir Y. Levy, Aurelien Lucchi, and Thomas Hofmann (2021). "Generative minimization networks: training GANs without competition". In: *arXiv:2103.12685*.

Grnarova, Paulina, Kfir Y. Levy, Aurelien Lucchi, Nathanael Perraudin, Ian Goodfellow, Thomas Hofmann, and Andreas Krause (2019). "A domain agnostic measure for monitoring and evaluating GANs". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Gross, Oliver Alfred and R. A. Wagner (1950). *A continuous Colonel Blotto game*. RAND Corporation.

Grover, Aditya, Eric Wang, Aaron Zweig, and Stefano Ermon (2019). "Stochastic optimization of sorting networks via continuous relaxations". In: *International Conference on Learning Representations*.

Guo, Xin, Anran Hu, Renyuan Xu, and Junzi Zhang (2019). "Learning mean-field games". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Gürkan, Gül and Jong-Shi Pang (2009). "Approximations of Nash equilibria". In: *Mathematical Programming*.

Ha, David, Andrew M. Dai, and Quoc V. Le (2017). "HyperNetworks". In: *International Conference on Learning Representations (ICLR)*.

Hairer, Ernst, Syvert P. Nørsett, and Gerhard Wanner (1993). *Solving ordinary differential equations I: nonstiff problems*. Springer-Verlag.

Hall, J., I. Galabova, L. Gottwald, and M. Feldmeier (2023). *HiGHS–high performance software for linear optimization*.

Hamilton, William D. (1964). "The genetical evolution of social behaviour. I and II". In: *Journal of Theoretical Biology*.

Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant (2020). "Array programming with NumPy". In: *Nature*.

Harsanyi, John (1967). "Games with incomplete information played by Bayesian players". In: *Management Science*.

Hastings, Wilfred (1970). "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification". In: *International Conference on Computer Vision (ICCV)*.

Heek, Jonathan, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee (2023). *Flax: a neural network library and ecosystem for JAX*.

Heinrich, Johannes, Marc Lanctot, and David Silver (2015). "Fictitious self-play in extensive-form games". In: *International Conference on Machine Learning (ICML)*.

Heinrich, Johannes and David Silver (2016). "Deep reinforcement learning from self-play in imperfect-information games". In: *arXiv:1603.01121*.

Heusinger, Anna Von and Christian Kanzow (2009a). "Optimization reformulations of the generalized Nash equilibrium problem using Nikaido–Isoda-type functions". In: *Computational Optimization and Applications*.

— (2009b). "Relaxation methods for generalized Nash equilibrium problems with inexact line search". In: *Journal of Optimization Theory and Applications (JOTA)*.

Hoffman, Matthew D. and Andrew Gelman (2014). "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research (JMLR)*.

Hornik, Kurt (1991). "Approximation capabilities of multilayer feedforward networks". In: *Neural networks*.

Hou, Jian, Zong-Chuan Wen, and Qing Chang (2018). "An unconstrained optimization reformulation for the Nash game". In: *Journal of Interdisciplinary Mathematics (JIM)*.

Hsieh, Yu-Guan, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos (2019). "On the convergence of single-call stochastic extra-gradient methods". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Hsieh, Ya-Ping, Panayotis Mertikopoulos, and Volkan Cevher (2021). "The limits of min-max optimization algorithms: convergence to spurious non-critical sets". In: *International Conference on Machine Learning (ICML)*.

Hu, Haigen, Xiaoyuan Wang, Yan Zhang, Qi Chen, and Qiu Guan (2024). "A comprehensive survey on contrastive learning". In: *Neurocomputing*.

Hu, Yaohua, Xiaoqi Yang, and Chee-Khian Sim (2015). "Inexact subgradient methods for quasi-convex optimization problems". In: *European Journal of Operational Research*.

Huang, Chin-Wei, David Krueger, Alexandre Lacoste, and Aaron Courville (2018). "Neural autoregressive flows". In: *International Conference on Machine Learning (ICML)*.

Huang, Kevin XD and Jan Werner (2000). "Asset price bubbles in Arrow–Debreu and sequential equilibrium". In: *Economic Theory*.

Huang, Minyi, Peter E. Caines, and Roland P. Malhamé (2007). "Large-population cost-coupled LQG problems with nonuniform agents: individual-mass behavior and decentralized epsilon-Nash equilibria". In: *IEEE Transactions on Automatic Control*.

Huang, Minyi, Roland P. Malhamé, and Peter E. Caines (2006). "Large population stochastic dynamic games: closed-loop McKean–Vlasov systems and the Nash certainty equivalence principle". In: *Communications in Information and Systems*.

Huangfu, Qi and J. A. Julian Hall (2018). "Parallelizing the dual revised simplex method". In: *Mathematical Programming Computation*.

Hunter, John (2007). "Matplotlib: a 2D graphics environment". In: *Computing in Science & Engineering*.

Isaacs, Rufus (1965). *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. John Wiley and Sons.

Ising, Ernst (1925). "Beitrag zur theorie des ferromagnetismus". In: *Zeitschrift für Physik*.

Itô, Kiyosi (1944). "Stochastic integral". In: *Proceedings of the Imperial Academy*.

Jaiswal, Ashish, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon (2020). "A survey on contrastive self-supervised learning". In: *Technologies*.

Ji, Kaiyi, Junjie Yang, and Yingbin Liang (2021). "Bilevel optimization: Convergence analysis and enhanced design". In: *International conference on machine learning*. Proceedings of Machine Learning Research (PMLR).

Jin, Chi, Praneeth Netrapalli, and Michael I. Jordan (2020). "What is local optimality in nonconvex-nonconcave minimax optimization?" In: *International Conference on Machine Learning (ICML)*.
Jonker, Roy and Ton Volgenant (1988). "A shortest augmenting path algorithm for dense and sparse linear assignment problems". In: *DGOR/NSOR*.
Jordan, James S. (1993). "Three problems in learning mixed-strategy Nash equilibria". In: *Games and Economic Behavior (GEB)*.
Kagel, John H. and Dan Levin (1993). "Independent private value auctions: bidder behaviour in first-, second- and third-price auctions with varying numbers of bidders". In: *The Economic Journal*.
Kakutani, Shizuo (1941). "A generalization of Brouwer's fixed point theorem". In: *Duke Mathematical Journal*.
Kamra, Nitin, Fei Fang, Debarun Kar, Yan Liu, and Milind Tambe (2017). "Handling continuous space security games with neural networks". In: *International Joint Conference on Artificial Intelligence (IJCAI)*.
Kamra, Nitin, Umang Gupta, Fei Fang, Yan Liu, and Milind Tambe (2018). "Policy learning for continuous space security games using neural networks". In: *AAAI Conference on Artificial Intelligence*.
Kamra, Nitin, Umang Gupta, Kai Wang, Fei Fang, Yan Liu, and Milind Tambe (2019). "DeepFP for finding Nash equilibrium in continuous action spaces". In: *Decision and Game Theory for Security*.
Kar, Debarun, Thanh H. Nguyen, Fei Fang, Matthew Brown, Arunesh Sinha, Milind Tambe, and Albert Xin Jiang (2017). "Trends and applications in Stackelberg security games". In: *Handbook of dynamic game theory*. Springer International Publishing.
Kessler, Donald J. and Burton G. Cour-Palais (1978). "Collision frequency of artificial satellites: the creation of a debris belt". In: *Journal of Geophysical Research: Space Physics*.
Le-Khac, Phuc, Graham Healy, and Alan Smeaton (2020). "Contrastive representation learning: a framework and review". In: *IEEE Access*.
Khan, Ali (1985). "Equilibrium points of nonatomic games over a nonreflexive Banach space". In: *Journal of Approximation Theory*.
— (1986). "Equilibrium points of nonatomic games over a Banach space". In: *Transactions of the American Mathematical Society*.
Khan, Ali and Nikolaos Papageorgiou (1987a). "On Cournot–Nash equilibria in generalized qualitative games with a continuum of players". In: *Nonlinear Analysis: Theory, Methods & Applications*.
— (1987b). "On Cournot–Nash equilibria in generalized qualitative games with an atomless measure space of agents". In: *Proceedings of the American Mathematical Society*.
Khan, Ali and Yeneng Sun (2002). "Non-cooperative games with many players". In: *Handbook of Game Theory with Economic Applications*.
Kidger, Patrick (2021). "On neural differential equations". PhD thesis. University of Oxford.
Kidger, Patrick and Terry Lyons (2020). "Universal approximation with deep narrow networks". In: *Conference on Learning Theory (COLT)*.
Kim, Taesung, Karel Prikry, and Nicholas Yannelis (1989). "Equilibria in abstract economies with a measure space of agents and with an infinite dimensional strategy space". In: *Journal of Approximation Theory*.
Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis,

Claudia Clopath, Dharshan Kumaran, and Raia Hadsell (2017). "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the National Academy of Sciences (PNAS)*.
Kiwiel, Krzysztof C. (2004). "Convergence of approximate and incremental subgradient methods for convex optimization". In: *SIAM Journal on Optimization*.
Kizilkale, A. Căağatay, Rabih Salhab, and Roland P. Malhamé (2019). "An integral control formulation of mean field game based large scale coordination of loads in smart grids". In: *Automatica*.
Kleinert, Thomas, Martine Labbé, Ivana Ljubić, and Martin Schmidt (2021). "A survey on mixed-integer programming techniques in bilevel optimization". In: *EURO Journal on Computational Optimization*.
Knuth, Donald E. (1976). "Big omicron and big omega and big theta". In: *ACM Sigact News*.
Koranne, Sandeep (2010). "Hierarchical data format 5: HDF5". In: *Handbook of open source tools*. Springer.
Korpelevich, Galina M. (1976). "The extragradient method for finding saddle points and other problems". In: *Ekonomika i Matematicheskie Metody*.
Korzhyk, Dmytro, Zhengyu Yin, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe (2011). "Stackelberg vs. Nash in security games". In: *Journal of Artificial Intelligence Research (JAIR)*.
Kovenock, Dan and Brian Roberson (2011). "A Blotto game with multi-dimensional incomplete information". In: *Economics Letters*.
— (2021). "Generalizations of the General Lotto and Colonel Blotto games". In: *Economic Theory*.
Krawczyk, Jacek B. (2005). "Coupled constraint Nash equilibria in environmental games". In: *Resource and Energy Economics*.
Krawczyk, Jacek B. and Stanislav Uryasev (2000). "Relaxation algorithms to find Nash equilibria with economic applications". In: *Environmental Modeling & Assessment*.
Krishna, Vijay (2002). *Auction theory*. Academic Press.
Kroer, Christian and Tuomas Sandholm (2015). "Discretization of continuous action spaces in extensive-form games". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
Kroupa, Tomáš and Tomáš Votroubek (2023). "Multiple oracle algorithm to solve continuous games". In: *Decision and Game Theory for Security*.
Kuhn, Harold W. (1955). "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly*.
Kuhn, Harold William (1950). "A simplified two-person poker". In: *Contributions to the Theory of Games*. Princeton University Press.
Kuipers, Lauwerens and Harald Niederreiter (1974). *Uniform distribution of sequences*. Wiley-Interscience.
Kurzweil, Jaroslav (1957). "Generalized ordinary differential equations and continuous dependence on a parameter". In: *Czechoslovak Mathematical Journal*.
Lanctot, Marc, Viliam Lisý, and Mark H. M. Winands (2014). "Monte Carlo tree search in simultaneous move games with applications to Goofspiel". In: *Computer Games*.
Lanctot, Marc, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Jason Upchurch, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Sutherland, Chris Anderson, Ulrich Siegler, Sandra Bechtle, Sarah McAleer, Richard Dehouck, David Kasenberg, Michael Ostercamp, Jordan Sahli, Victor Shkurti, Venugopal Lakshminarayanan, Panagiotis Kougiouris, Justin Fu, Houghton

Bell, Abhishek Sen, and Stephen Knight (2019). "OpenSpiel: a framework for reinforcement learning in games". In: *arXiv:1908.09453*.
Lanctot, Marc, 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: *Conference on Neural Information Processing Systems (NeurIPS)*.
Landau, Edmund (1909). *Handbuch der Lehre von der Verteilung der Primazahlen (Handbook of the Theory of the Distribution of Prime Numbers)*. BG Teubner.
Lasry, Jean-Michel and Pierre-Louis Lions (2007). "Mean field games". In: *Japanese Journal of Mathematics*.
Laurière, Mathieu, Sarah Perrin, Julien Pérolat, Sertan Girgin, Paul Muller, Romuald Élie, Matthieu Geist, and Olivier Pietquin (2022). "Learning mean field games: a survey". In: *arXiv:2205.12944*.
Lebesgue, Henri (1902). "Intégrale, longueur, aire". In: *Annali di Matematica Pura ed Applicata (1898-1922)*.
LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *Nature*.
Lenc, Karel, Erich Elsen, Tom Schaul, and Karen Simonyan (2019). "Non-differentiable supervised learning with evolution strategies and hybrid methods". In: *arXiv:1906.03139*.
Lenz, Wilhelm (1920). "Beitrag zum Verständnis der magnetischen Erscheinungen in festen Körpern". In: *Physikalische Zeitschrift*.
Leonidov, Andrey, Alexey Savvateev, and Andrew Semenov (2020). "QRE in the Ising game". In: *CEUR Workshop*.
— (2024). "Ising game on graphs". In: *Chaos, Solitons & Fractals*.
Leshno, Moshe, Vladimir Lin, Allan Pinkus, and Shimon Schocken (1993). "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural networks*.
Leslie, David S. and Edmund J. Collins (2003). "Convergent multiple-timescales reinforcement learning algorithms in normal form games". In: *The Annals of Applied Probability*.
— (2006). "Generalised weakened fictitious play". In: *Games and Economic Behavior (GEB)*.
Letcher, Alistair (2018). "Stability and exploitation in differentiable games". MA thesis. University of Oxford.
Letcher, Alistair, David Balduzzi, Sébastien Racanière, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel (2019). "Differentiable game mechanics". In: *Journal of Machine Learning Research (JMLR)*.
Letcher, Alistair, Jakob Foerster, David Balduzzi, Tim Rocktäschel, and Shimon Whiteson (2019). "Stable opponent shaping in differentiable games". In: *International Conference on Learning Representations (ICLR)*.
Levine, David K. (1998). "Modeling altruism and spitefulness in experiments". In: *Review of Economic Dynamics*.
Li, Zhizhong and Derek Hoiem (2017). "Learning without forgetting". In: *IEEE transactions on pattern analysis and machine intelligence*.
Li, Zun and Michael P. Wellman (2021). "Evolution strategies for approximate solution of Bayesian games". In: *AAAI Conference on Artificial Intelligence*.
Lin, Henry W., Max Tegmark, and David Rolnick (2017). "Why does deep and cheap learning work so well?" In: *Journal of Statistical Physics*.
Lisý, Viliam and Michael Bowling (2017). "Equilibrium approximation quality of current no-limit poker bots". In: *AAAI Computer Poker Workshop*.

Lockhart, Edward, 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: *International Joint Conference on Artificial Intelligence (IJCAI)*.

Loizou, Nicolas, Hugo Berard, Gauthier Gidel, Ioannis Mitliagkas, and Simon Lacoste-Julien (2021). "Stochastic gradient descent-ascent and consensus optimization for smooth games: convergence analysis under expected co-coercivity". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Lorraine, Jonathan and David Duvenaud (2018). "Stochastic hyperparameter optimization through hypernetworks". In: *arXiv:1802.09419*.

Lorraine, Jonathan, Paul Vicol, and David Duvenaud (2020). "Optimizing millions of hyperparameters by implicit differentiation". In: *International conference on artificial intelligence and statistics*. Proceedings of Machine Learning Research (PMLR).

Lotker, Zvi, Boaz Patt-Shamir, and Mark R. Tuttle (2008). "A game of timing and visibility". In: *Games and Economic Behavior (GEB)*.

Lu, Zhou, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang (2017). "The expressive power of neural networks: a view from the width". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Luce, R Duncan (1959). *Individual choice behavior*. Wiley New York.

— (1977). "The choice axiom after twenty years". In: *Journal of mathematical psychology*.

Ma, Jeffrey, Alistair Letcher, Florian Schäfer, Yuanyuan Shi, and Anima Anandkumar (2021). "Polymatrix competitive gradient descent". In: *arXiv:2111.08565*.

Macal, C. M. and M. J. North (2010). "Tutorial on agent-based modelling and simulation". In: *Journal of Simulation*.

MacKay, Matthew, Paul Vicol, Jon Lorraine, David Duvenaud, and Roger Grosse (2019). "Self-tuning networks: bilevel optimization of hyperparameters using structured best-response functions". In: *arXiv:1903.03088*.

Marchesi, Alberto, Francesco Trovò, and Nicola Gatti (2020). "Learning probably approximately correct maximin strategies in simulation-based games with infinite strategy spaces". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Marris, Luke, Paul Muller, Marc Lanctot, Karl Tuyls, and Thore Graepel (2021). "Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers". In: *International Conference on Machine Learning (ICML)*.

Martin, Carlos, Craig Boutilier, Ofer Meshi, and Tuomas Sandholm (2024). "Model-free preference elicitation". In: *International Joint Conference on Artificial Intelligence (IJCAI)*.

Martin, Carlos and Tuomas Sandholm (2021a). "Bayesian multiagent inverse reinforcement learning for policy recommendation". In: *AAAI Conference on Artificial Intelligence Workshop on Reinforcement Learning in Games*.

— (2021b). "Efficient exploration of zero-sum stochastic games". In: *AAAI Conference on Artificial Intelligence Workshop on Reinforcement Learning in Games*.

— (2023). "Finding mixed-strategy equilibria of continuous-action games without gradients using randomized policy networks". In: *International Joint Conference on Artificial Intelligence (IJCAI)*.

— (2025a). "AlphaZeroES: direct score maximization outperforms planning loss minimization". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

— (2025b). "ApproxED: approximate exploitability descent via learned best responses". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Martin, Carlos and Tuomas Sandholm (2025c). "Incremental multiple oracle". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
— (2025d). "Joint-perturbation simultaneous pseudo-gradient". In: *International Joint Conference on Artificial Intelligence (IJCAI)*.
— (2025e). "Solving infinite-player games with player-to-strategy networks". In: *arXiv:2501.09330*.
Mazumdar, Eric, Lillian J. Ratliff, and S. Shankar Sastry (2020). "On gradient-based learning in continuous games". In: *SIAM Journal on Mathematics of Data Science (SIMODS)*.
Mazumdar, Eric, S. Shankar Sastry, and Michael I. Jordan (2025). "On finding local Nash equilibria (and only local Nash equilibria) in zero-sum games". In: *ACM/IMS Journal of Data Science*.
McAleer, Stephen, JB Lanier, Kevin A. Wang, Pierre Baldi, Tuomas Sandholm, and Roy Fox (2024). "Toward optimal policy population growth in two-player zero-sum games". In: *International Conference on Learning Representations (ICLR)*.
McAleer, Stephen, John B. Lanier, Roy Fox, and Pierre Baldi (2020). "Pipeline PSRO: a scalable approach for finding approximate Nash equilibria in large games". In: *Conference on Neural Information Processing Systems (NeurIPS)*.
McAleer, Stephen, John B. Lanier, Kevin A. Wang, Pierre Baldi, and Roy Fox (2021). "XDO: a double oracle algorithm for extensive-form games". In: *Conference on Neural Information Processing Systems (NeurIPS)*.
McAleer, Stephen, Kevin Wang, John Lanier, Marc Lanctot, Pierre Baldi, Tuomas Sandholm, and Roy Fox (2022). "Anytime PSRO for two-player zero-sum games". In: *arXiv:2201.07700*.
McCloskey, Michael and Neal J. Cohen (1989). "Catastrophic interference in connectionist networks: the sequential learning problem". In: *Psychology of learning and motivation*.
McCulloch, Warren S. and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics*.
McKee, Kevin R., Ian Gemp, Brian McWilliams, Edgar A. Duèñez-Guzmán, Edward Hughes, and Joel Z. Leibo (2020). "Social diversity and social preferences in mixed-motive reinforcement learning". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
McKelvey, Richard D. and Thomas R. Palfrey (1995). "Quantal response equilibria for normal form games". In: *Games and economic behavior*.
— (1998). "Quantal response equilibria for extensive form games". In: *Experimental economics*.
McKinney, Wes (2010). "Data structures for statistical computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman.
McMahan, H. Brendan, Geoffrey J. Gordon, and Avrim Blum (2003). "Planning in the presence of cost functions controlled by an adversary". In: *International Conference on Machine Learning (ICML)*.
McNamara, John M. and Alasdair I. Houston (1996). "State-dependent life histories". In: *Nature*.
Mertikopoulos, Panayotis and Mathias Staudigl (2018). "On the convergence of gradient-like flows with noisy gradient input". In: *SIAM Journal on Optimization*.
Mertikopoulos, Panayotis and Zhengyuan Zhou (2019). "Learning in games with continuous action sets and unknown payoff functions". In: *Mathematical Programming*.
Merz, Antony W. (1974). "The homicidal chauffeur". In: *AIAA Journal*.
Mescheder, Lars, Sebastian Nowozin, and Andreas Geiger (2017). "The numerics of GANs". In: *Conference on Neural Information Processing Systems (NeurIPS)*.
Metropolis, Nicholas, Arianna Rosenbluth, Marshall Rosenbluth, Augusta Teller, and Edward Teller (1953). "Equation of state calculations by fast computing machines". In: *The Journal of Chemical Physics*.

Metz, Luke, Ben Poole, David Pfau, and Jascha Sohl-Dickstein (2017). "Unrolled generative adversarial networks". In: *International Conference on Learning Representations (ICLR)*.
Milchtaich, Igal (1996). *Generic uniqueness of equilibria in nonatomic congestion games*. Hebrew University of Jerusalem.
— (2000). "Generic uniqueness of equilibrium in large crowding games". In: *Mathematics of Operations Research*.
— (2005). "Topological conditions for uniqueness of equilibrium in networks". In: *Mathematics of Operations Research*.
Milgrom, Paul and Ilya Segal (2014). "Deferred-acceptance auctions and radio spectrum reallocation". In: *ACM Conference on Economics and Computation (EC)*.
— (2020). "Clock auctions and radio spectrum reallocation". In: *Journal of Political Economy*.
Milgrom, Paul and Robert Weber (1985). "Distributional strategies for games with incomplete information". In: *Mathematics of Operations Research*.
Mitchell, Tom (1997). *Machine Learning*. McGraw-Hill.
Mobiny, Aryan, Pengyu Yuan, Supratik K. Moulik, Naveen Garg, Carol C. Wu, and Hien Van Nguyen (2021). "DropConnect is effective in modeling uncertainty of Bayesian deep networks". In: *Scientific reports*.
Monderer, Dov and Lloyd S. Shapley (1996). "Potential games". In: *Games and Economic Behavior (GEB)*.
Moravčík, Matej, 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". In: *Science*.
Morgan, John, Kenneth Steiglitz, and George Reis (2003). "The spite motive and equilibrium behavior in auctions". In: *Contributions to Economic Analysis & Policy*.
Moulin, H. and J.-P. Vial (1978). "Strategically zero-sum games: the class of games whose completely mixed equilibria cannot be improved upon". In: *International Journal of Game Theory*.
Muller, Paul, Shayegan Omidshafiei, Mark Rowland, Karl Tuyls, Julien Perolat, Siqi Liu, Daniel Hennes, Luke Marris, Marc Lanctot, Edward Hughes, Zhe Wang, Guy Lever, Nicolas Heess, Thore Graepel, and Remi Munos (2020). "A generalized training approach for multiagent learning". In: *International Conference on Learning Representations (ICLR)*.
Muller, Paul, Mark Rowland, Romuald Elie, Georgios Piliouras, Julien Perolat, Mathieu Lauriere, Raphael Marinier, Olivier Pietquin, and Karl Tuyls (2022). "Learning equilibria in mean-field games: introducing mean-field PSRO". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
Myerson, Roger B. (1991). *Game theory: analysis of Conflict*. Harvard University Press.
Nash, John F. (1950). "Equilibrium points in n-person games". In: *Proceedings of the National Academy of Sciences (PNAS)*.
— (1951). "Non-cooperative games". In: *Annals of Mathematics*.
Neal, Radford M. (1996). "Monte Carlo implementation". In: *Bayesian learning for neural networks*. Springer.
Nesterov, Yurii and Vladimir Spokoiny (2017). "Random gradient-free minimization of convex functions". In: *Foundations of Computational Mathematics*.
Niederreiter, Harald (1992). *Random number generation and quasi-Monte Carlo methods*. SIAM.
Nikaido, Hukukane and Kazuo Isoda (1955). "Note on non-cooperative convex games". In: *Pacific Journal of Mathematics*.

Novshek, William (1985). "Perfectly competitive markets as the limits of Cournot markets". In: *Journal of Economic Theory (JET)*.
Omidshafiei, Shayegan, Christos Papadimitriou, Georgios Piliouras, Karl Tuyls, Mark Rowland, Jean-Baptiste Lespiau, Wojciech Czarnecki, Marc Lanctot, Julien Perolat, and Remi Munos (2019). "α-rank: multi-agent evaluation by evolution". In: *Scientific Reports*.
Osborne, Martin J. (2004). *An introduction to game theory*. Oxford University Press.
Padala, Manisha, Debojit Das, and Sujit Gujar (2021). "Effect of input noise dimension in GANs". In: *Neural Information Processing*.
Parise, Francesca and Asuman Ozdaglar (2019). "Graphon games". In: *Economics and Computation*.
— (2023). "Graphon games: a statistical framework for network games and interventions". In: *Econometrica*.
Patsko, Valery S. and Varvara L. Turova (2001). "Level sets of the value function in differential games with the homicidal chauffeur dynamics". In: *International Game Theory Review*.
Perkins, Steven and David S. Leslie (2014). "Stochastic fictitious play with continuous action sets". In: *Journal of Economic Theory (JET)*.
Perolat, Julien, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean-Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Remi Munos, David Silver, Satinder Singh, Demis Hassabis, and Karl Tuyls (2022). "Mastering the game of Stratego with model-free multiagent reinforcement learning". In: *Science*.
Pérolat, Julien, Sarah Perrin, Romuald Elie, Mathieu Laurière, Georgios Piliouras, Matthieu Geist, Karl Tuyls, and Olivier Pietquin (2022). "Scaling mean field games by online mirror descent". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
Perrin, Sarah, Julien Perolat, Mathieu Laurière, Matthieu Geist, Romuald Elie, and Olivier Pietquin (2020). "Fictitious play for mean field games: continuous time analysis and applications". In: *Conference on Neural Information Processing Systems (NeurIPS)*.
Petersen, Peter (2006). *Riemannian geometry*. Springer.
Pettis, Billy James (1938). "On integration in vector spaces". In: *Transactions of the American Mathematical Society*.
Pillards, Tim and Ronald Cools (2005). "Transforming low-discrepancy sequences from a cube to a simplex". In: *Journal of computational and applied mathematics*.
Pinkus, Allan (1999). "Approximation theory of the MLP model in neural networks". In: *Acta numerica*.
Pitis, Silviu (2019). "Rethinking the discount factor in reinforcement learning: a decision theoretic approach". In: *AAAI Conference on Artificial Intelligence*.
Plackett, Robin L. (1975). "The analysis of permutations". In: *Journal of the Royal Statistical Society Series C: Applied Statistics*.
Poisson, Siméon-Denis (1837). *Recherches sur la probabilité des jugements en matière criminelle et en matière civile: précédées des règles générales du calcul des probabilités*. Bachelier.
Popov, Leonid (1980). "A modification of the Arrow–Hurwicz method for search of saddle points". In: *Mathematical Notes*.
Qin, Rong-Jun, Fan-Ming Luo, Hong Qian, and Yang Yu (2022). "Unified policy optimization for continuous-action reinforcement learning in non-stationary tasks and games". In: *arXiv:2208.09452*.

Qu, Biao and Jing Zhao (2013). "Methods for solving generalized Nash equilibrium". In: *Journal of Applied Mathematics*.

Raghunathan, Arvind, Anoop Cherian, and Devesh Jha (2019). "Game theoretic optimization via gradient-based Nikaido–Isoda function". In: *International Conference on Machine Learning (ICML)*.

Rahaman, Nasim, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville (2019). "On the spectral bias of neural networks". In: *International Conference on Machine Learning (ICML)*.

Rath, Kali (1992). "A direct proof of the existence of pure strategy equilibria in games with a continuum of players". In: *Economic Theory*.

Ray, Debraj and Rajiv Vohra (2020). "Games of love and hate". In: *Journal of Political Economy*.

Rebuffi, Sylvestre-Alvise, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert (2017). "iCaRL: incremental classifier and representation learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rechenberg, Ingo (1973). *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog.

— (1978). "Evolutionsstrategien". In: *Simulationsmethoden in der medizin und biologie*. Springer.

Reluga, Timothy C. (2010). "Game theory of social distancing in response to an epidemic". In: *PLoS Computational Biology*.

Rhoads, Glenn C. and Laurent Bartholdi (2012). "Computer solution to the game of pure strategy". In: *Games*.

Riemann, Bernhard (1868). "Über die Darstellbarkeit einer Function durch eine trigonometrische Reihe". In: *Abhandlungen der Königlichen Gesellschaft der Wissenschaften in Göttingen*.

Roberts, Martin (2018). *The unreasonable effectiveness of quasirandom sequences*.

Rosen, J. Ben (1965). "Existence and uniqueness of equilibrium points for concave n-person games". In: *Econometrica*.

Rosenblatt, Frank (1958). "The perceptron: a probabilistic model for information storage and organization in the brain". In: *Psychological review*.

Ross, Sheldon M. (1971). "Goofspiel: the game of pure strategy". In: *Journal of Applied Probability*.

Rossum, Guido Van and Fred Drake Jr (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Rusu, Andrei A., Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell (2016). "Progressive neural networks". In: *arXiv:1606.04671*.

Saijo, Tatsuyoshi and Hideki Nakamura (1995). "The "spite" dilemma in voluntary contribution mechanism experiments". In: *Journal of Conflict Resolution*.

Saks, Stanisław (1937). *Theory of the Integral*. Monografie matematyczne.

Salimans, Tim, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever (2017). "Evolution strategies as a scalable alternative to reinforcement learning". In: *arXiv:1703.03864*.

Sandholm, Tuomas (2013). "Very-large-scale generalized combinatorial multi-attribute auctions: lessons from conducting $60 billion of sourcing". In: *Handbook of Market Design*. Oxford University Press.

Sandholm, William H. (2001). "Potential games with continuous player sets". In: *Journal of Economic Theory*.

Santos, Manuel S. and Michael Woodford (1997). "Rational asset pricing bubbles". In: *Econometrica: Journal of the Econometric Society*.

Sard, Arthur (1942). "The measure of the critical values of differentiable maps". In: *Bulletin of the American Mathematical Society*.
Schäfer, Florian and Anima Anandkumar (2019). "Competitive gradient descent". In: *Conference on Neural Information Processing Systems (NeurIPS)*.
Schmeidler, David (1973). "Equilibrium points of nonatomic games". In: *Journal of Statistical Physics*.
Schmidhuber, Jürgen (1992). "Learning to control fast-weight memories: an alternative to dynamic recurrent networks". In: *Neural Computation*.
Schwarz, Hermann Amandus (1890). "Ueber ein die Flächen kleinsten Flächeninhalts betreffendes Problem der Variationsrechnung: Festschrift zum siebzigsten Geburtstage des Herrn Karl Weierstrass". In: *Gesammelte Mathematische Abhandlungen: Erster Band*. Springer.
Schwefel, Hans-Paul (1977). *Numerische optimierung von computer-modellen mittels der evolutionsstrategie*. Birkhäuser Basel.
Selten, Reinhard (1970). *Preispolitik der Mehrproduktenunternehmung in der statischen Theorie*. Springer Berlin, Heidelberg.
Shamir, Ohad (2017). "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback". In: *Journal of Machine Learning Research (JMLR)*.
Shapley, Lloyd S. (1964). "Some topics in two-person games". In: *Advances in Game Theory*. Princeton University Press.
Shapley, Lloyd S. and Martin Shubik (1971). "The assignment game I: the core". In: *International Journal of Game Theory*.
Sharma, Ankit and Tuomas Sandholm (2010). "Asymmetric spite in auctions". In: *AAAI Conference on Artificial Intelligence*.
Shin, Hanul, Jung Kwon Lee, Jaehong Kim, and Jihoon Kim (2017). "Continual learning with deep generative replay". In: *Conference on Neural Information Processing Systems (NeurIPS)*.
Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis (2018). "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play". In: *Science*.
Simon, Herbert A. (1955). "A behavioral model of rational choice". In: *The Quarterly Journal of Economics*.
Sinha, Arunesh, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe (2018). "Stackelberg security games: looking beyond a decade of success". In: *International Joint Conference on Artificial Intelligence (IJCAI)*.
Sion, Maurice (1958). "On general minimax theorems". In: *Pacific Journal of Mathematics*.
Smith, John Maynard (1974). "The theory of games and evolution in animal conflict". In: *Journal of Theoretical Biology*.
Spall, James C. (1992). "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation". In: *IEEE Transactions on Automatic Control*.
— (1997). "A one-measurement form of simultaneous perturbation stochastic approximation". In: *Automatica*.
Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *Journal of Machine Learning Research (JMLR)*.
Steinberger, Eric, Adam Lerer, and Noam Brown (2020). "DREAM: deep regret minimization with advantage baselines and model-free learning". In: *arXiv:2006.10410*.

Stieltjes, Thomas Joannes (1894). "Recherches sur les fractions continues". In: *Annales de la Faculté des sciences de Toulouse: Mathématiques*.
Sutton, Richard S. and Andrew G. Barto (1998). *Reinforcement learning: an introduction*. MIT press Cambridge.
Szafron, Duane, Richard G. Gibson, and Nathan R. Sturtevant (2013). "A parameterized family of equilibrium profiles for three-player Kuhn poker". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
Szentes, Balázs and Robert W. Rosenthal (2003a). "Beyond chopsticks: symmetric equilibria in majority auction games". In: *Games and Economic Behavior*.
— (2003b). "Three-object two-bidder simultaneous auctions: chopsticks and tetrahedra". In: *Games and Economic Behavior (GEB)*.
Talvila, Erik (2001). "Necessary and sufficient conditions for differentiating under the integral sign". In: *The American Mathematical Monthly*.
Tancik, Matthew, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng (2020). "Fourier features let networks learn high frequency functions in low dimensional domains". In: *Conference on Neural Information Processing Systems (NeurIPS)*.
Tang, Pingzhong and Tuomas Sandholm (2012). "Optimal auctions for spiteful bidders". In: *AAAI Conference on Artificial Intelligence*.
Telgarsky, Matus (2016). "Benefits of depth in neural networks". In: *Conference on Learning Theory (COLT)*.
The HDF Group (2024). *Hierarchical data format, version 5*. DOI: 10.5281/zenodo.17808614.
The Pandas development team (2020). *Pandas*. DOI: 10.5281/zenodo.3509134.
Timbers, Finbarr, Nolan Bard, Edward Lockhart, Marc Lanctot, Martin Schmid, Neil Burch, Julian Schrittwieser, Thomas Hubert, and Michael Bowling (2022). "Approximate exploitability: learning a best response". In: *International Joint Conference on Artificial Intelligence (IJCAI)*.
Tsaknakis, Ioannis and Mingyi Hong (2021). "Finding first-order Nash equilibria of zero-sum games with the regularized Nikaido–Isoda function". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*.
Tucker, Albert W. (1984). *The Princeton mathematics community in the 1930s: transcript number 11*.
Uryasev, Stanislav and Reuven Y. Rubinstein (1994). "On relaxation algorithms in computation of noncooperative equilibria". In: *IEEE Transactions on Automatic Control (TACON)*.
Vickrey, William S. (1969). "Congestion theory and transport investment". In: *American Economic Review (AER)*.
Vinyals, Oriol, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver (2019). "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature*.
Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der

Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors (2020). "SciPy 1.0: fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods*.

Vitter, Jeffrey S. (1985). "Random sampling with a reservoir". In: *ACM Transactions on Mathematical Software (TOMS)*.

Vlatakis-Gkaragkounis, Emmanouil-Vasileios, Lampros Flokas, Thanasis Lianeas, Panayotis Mertikopoulos, and Georgios Piliouras (2020). "No-regret learning and mixed nash equilibria: They do not mix". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

von Neumann, John (1928). "Zur theorie der gesellschaftsspiele". In: *Mathematische Annalen*.

von Neumann, John and Oskar Morgenstern (1947). *Theory of games and economic behavior*. Princeton University Press.

Voorhees, Alan M. (1955). "A general theory of traffic movement". In: *Institute of Traffic Engineers*.

Walton, Michael and Viliam Lisy (2021). "Multi-agent reinforcement learning in OpenSpiel: a reproduction report". In: *arXiv:2103.00187*.

Wan, Li, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus (2013). "Regularization of neural networks using DropConnect". In: *International Conference on Machine Learning (ICML)*. Proceedings of Machine Learning Research (PMLR).

Wang, Yongzhao and Michael Wellman (2023). "Empirical game-theoretic analysis for mean field games". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Wang, Yuanhao, Guodong Zhang, and Jimmy Ba (2020). "On solving minimax optimization locally: a follow-the-ridge approach". In: *International Conference on Learning Representations (ICLR)*.

Wardrop, John Glen (1952). "Some theoretical aspects of road traffic research". In: *Proceedings of the Institution of Civil Engineers*.

Washburn, Alan (2013). "OR forum - Blotto politics". In: *Operations Research*.

Wellman, Michael P., Karl Tuyls, and Amy Greenwald (2025). "Empirical game theoretic analysis: a survey". In: *Journal of Artificial Intelligence Research (JAIR)*.

White, Martha (2017). "Unifying task specification in reinforcement learning". In: *International Conference on Machine Learning (ICML)*.

Wiener, Norbert (1923). "Differential-space". In: *Journal of Mathematics and Physics*.

— (1938). "The homogeneous chaos". In: *American Journal of Mathematics*.

Wierstra, Daan, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber (2014). "Natural evolution strategies". In: *Journal of Machine Learning Research (JMLR)*.

Willi, Timon, Alistair Letcher, Johannes Treutlein, and Jakob Foerster (2022). "COLA: consistent learning with opponent-learning awareness". In: *International Conference on Machine Learning (ICML)*.

Wilson, Alan G. (1967). "A statistical theory of spatial distribution models". In: *Transportation Research*.

Wiszniewska-Matyszkiel, Agnieszka (2014). "Open and closed loop Nash equilibria in games with a continuum of players". In: *Journal of Optimization Theory and Applications*.

Worth, Carl and Keith Packard (2003). "Xr: cross-device rendering for vector graphics". In: *Proceedings of the Ottawa Linux Symposium (OLS)*.

Wu, Zida, Mathieu Laurière, Samuel Jia Cong Chua, Matthieu Geist, Olivier Pietquin, and Ankur Mehta (2024). "Population-aware Online Mirror Descent for Mean-Field Games by Deep Re-

inforcement Learning". In: *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Xie, Qiaomin, Zhuoran Yang, Zhaoran Wang, and Andreea Minca (2021). "Learning while playing in mean-field games: Convergence and optimality". In: *International Conference on Machine Learning (ICML)*.

Xin, Chen, G. Yang, and J. P. Huang (2017). "Ising game: nonequilibrium steady states of resource-allocation systems". In: *Physica A*.

Yang, Zhe and Qingping Song (2022). "A unified approach to the Nash equilibrium existence in large games from finitely many players to infinitely many players". In: *Journal of Fixed Point Theory and Applications*.

Yarotsky, Dmitry (2017). "Error bounds for approximations with deep ReLU networks". In: *Neural Networks*.

Yellott Jr, John I. (1977). "The relationship between Luce's choice axiom, Thurstone's theory of comparative judgment, and the double exponential distribution". In: *Journal of Mathematical Psychology*.

Yi, Sun, Daan Wierstra, Tom Schaul, and Jürgen Schmidhuber (2009). "Stochastic search using the natural gradient". In: *International Conference on Machine Learning (ICML)*.

Yosida, Kôsaku and Edwin Hewitt (1952). "Finitely additive measures". In: *Transactions of the American Mathematical Society*.

Zhang, Chenyu, Xu Chen, and Xuan Di (2025). "Stochastic semi-gradient descent for learning mean field games with population-aware function approximation". In: *International Conference on Learning Representations (ICLR)*.

Zhang, Chongjie and Victor Lesser (2010). "Multi-agent learning with policy prediction". In: *AAAI Conference on Artificial Intelligence*.

Zhuang, Juntang, Tommy Tang, Yifan Ding, Sekhar C. Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan (2020). "AdaBelief optimizer: adapting stepsizes by the belief in observed gradients". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Zinkevich, Martin, Michael Bowling, Michael Johanson, and Carmelo Piccione (2007). "Regret minimization in games with incomplete information". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Appendix A

Other completed work

In addition to the work presented in this thesis, I have also completed other projects during my PhD that, while valuable, fall outside the primary focus of this thesis project. These efforts demonstrate my broader research capabilities and collaborative experience.

- In Martin and Sandholm (2021b), we studied **efficient exploration of two-player zero-sum stochastic games** when the learner can control both players, only has oracle access via gameplay, and has a limited-duration exploration phase during which it can control both players. The objective is to output a strategy with low exploitability after exploration. We proposed using a belief distribution over possible environments to induce a distribution over state–action value functions and evaluate exploration strategies (including generalizations of Thompson sampling and Bayes-UCB), finding that Thompson-style and Bayes-UCB–style methods consistently outperform alternatives.
- In Martin and Sandholm (2021a), we studied **Bayesian multiagent inverse reinforcement learning for policy recommendation**. In this setting, an observer (the recommender) watches players' actions and state transitions in a known zero-sum game but not their rewards, and uses a Bayesian posterior (with behavioral models that may be Nash, quantal response, etc.) to infer players' reward functions. We proposed Bayesian inference procedures to recover posteriors over rewards/behaviors and show that the inferred preferences can be used to distinguish solution concepts and drive policy recommendations.
- In Martin, Boutilier, et al. (2024), we tackled **model-free preference elicitation** in recommender systems. Preference elicitation allows the system to ask a user questions to learn about their preferences and improve recommendation quality. We developed model-free variants of **expected value of information (EVOI)** that learn user-response and utility models from existing data (using function approximation) so preference-elicitation queries can be scored without explicit probabilistic preference models, and augment these learned models with online planning via Monte-Carlo tree search. Empirically, this approach yielded significant improvements in recommendation quality over standard baselines across several preference-elicitation tasks.
- Planning at execution time can dramatically improve performance in some settings. AlphaZero (Silver et al., 2018) is a state-of-the-art technique based on Monte Carlo Tree Search (MCTS) (Coulom, 2007). In Martin and Sandholm (2025a), we introduced **AlphaZeroES**. It modifies AlphaZero by replacing its loss with direct maximization of the episode score, while keeping

the MCTS algorithm and neural architecture unchanged. Since the episode score is generally not differentiable, we used evolution strategies (as described in Section 2.8) for this. Across several single-agent environments, experiments showed that directly maximizing episode score consistently and often dramatically outperforms minimizing the standard planning loss.

- In Martin and Sandholm (2025c), we presented **incremental multiple oracle** (IMO), a framework for computing approximate mixed-strategy Nash equilibria of continuous-action games. It is a modification of the traditional double oracle algorithm, extended to multiple players and continuous action spaces. It does not require exact metagame solving on each iteration, which can be computationally expensive for large metagames. Also, it does not require global best-response computation on each iteration, which can be computationally expensive or even intractable for high-dimensional action spaces and general games.

Appendix B

Additional information

Here, we include some additional information pertaining to the content presented in the body.

B.1 Additional information about auctions

Table B.1 describes the independent private values, common values, affiliated values, complete information, and asymmetric information auctions, in that order.

Ω	$\tau_i(\omega)$	$v_i(\omega)$
$[0,1]^n$	ω_i	ω_i
$[0,1]^{n+1}$	$\omega_i\omega_{n+1}$	ω_{n+1}
$[0,1]^{n+1}$	$\omega_i + \omega_{n+1}$	$\omega_{n+1} + \frac{1}{n}\sum_{i=1}^{n}\omega_i$
$[0,1]$	ω	ω
$[0,1]$	$\omega[\![i=1]\!]$	ω

Table B.1: Auction descriptions. All use $\mu = \mathcal{U}(\Omega)$.

For the common values auction, also known as the "mineral rights" model (Krishna, 2002, example 6.1), the following procedure can be used to sample $\omega \mid o_i$:

$$z \sim \mathcal{U}([0,1]) \tag{B.1}$$
$$\omega_{n+1} = o_i^z \tag{B.2}$$
$$\omega_i = o_i \,/\, \omega_{n+1} \tag{B.3}$$
$$\omega_j \sim \mathcal{U}([0,1]) \qquad j \neq i \tag{B.4}$$

For the affiliated values auction (Krishna, 2002, example 6.2), the following procedure can be used to sample $\omega \mid o_i$:

$$\omega_{n+1} \sim \mathcal{U}(\max\{0, o_i - 1\}, \min\{1, o_i\}) \tag{B.5}$$
$$\omega_i = o_i - \omega_{n+1} \tag{B.6}$$
$$\omega_j \sim \mathcal{U}([0,1]) \qquad j \neq i \tag{B.7}$$

The all-pay auction with independent private values has a pure symmetric equilibrium generated by:

$$a_i = \tfrac{n-1}{n}(o_i)^n \tag{B.8}$$

The kth-price winner-pay auction with independent private values has a pure symmetric equilibrium generated by Kagel and Levin (1993, p. 878):

$$a_i = \tfrac{n-1}{n+1-k} o_i \tag{B.9}$$

The 3-player 2nd-price winner-pay auction with common values has a pure symmetric equilibrium generated by Bichler, Fichtl, Heidekrüger, et al. (2021):

$$a_i = \tfrac{o_i}{2+\frac{1}{2}o_i} \tag{B.10}$$

The 2-player 1st- and 2nd-price winner-pay auction with affiliated values have pure symmetric equilibria generated by, respectively (Bichler, Fichtl, Heidekrüger, et al., 2021):

$$a_i = \tfrac{2}{3} o_i \tag{B.11}$$

$$a_i = o_i \tag{B.12}$$

B.2 Noise dimensionality in RPNs

For randomized policy networks (Section 5.1), the dimensionality of the input noise is crucial to performance. In this section, we review some results from the literature pertaining to this.

As described in Section 1.1, universal approximation theorems guarantee that neural networks can approximate arbitrary continuous functions on a compact domain.

Furthermore, if $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^m$, X is an $\mathcal{X}$-valued random variable, $f : \mathcal{X} \to \mathcal{Y}$, and f_n is a sequence of functions that converges pointwise to f, the sequence of random variables $Y_n = f_n(X)$ converges in distribution to $Y = f(X)$ (Huang, Krueger, et al., 2018, Lemma 4).

Gaussian input noise of *lower* dimension than the output space does not suffice to approximate arbitrary distributions on the output space. In particular, **Sard's theorem** (Sard, 1942; Petersen, 2006) says the following: Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be k times continuously differentiable, where $k \geq \max\{n - m + 1, 1\}$. Let $X \subseteq \mathbb{R}^n$ be the set of critical points of f (that is, points where the Jacobian matrix of f has rank less than m). Then the image $f[X]$ has Lebesgue measure zero in $\mathbb{R}^m$. As a corollary, if $n < m$, then all points in $\mathbb{R}^n$ are critical. Thus $\operatorname{im} f$ has Lebesgue measure zero in $\mathbb{R}^m$.[1]

Padala, Das, and Gujar (2021) study the effect of input noise dimension in GANs. They show that the right dimension of input noise for optimal results depends on the dataset and architecture used.

Feng, Zhao, and Zha (2021) study how noise injection in GANs helps them overcome the "adversarial dimension trap", which arises when the generated manifold has an intrinsic dimension lower than that of the data manifold: that is, when the latent space is low-dimensional compared to the high-dimensional space of real image details. Citing Sard's theorem, they advise against mapping low-dimensional feature spaces into feature manifolds with higher intrinsic dimensions.

[1] From a standard uniform random variable, one can extract two independent variables by de-interleaving its binary expansion, but this operation is highly discontinuous and pathological.

Bailey and Telgarsky (2018) investigate the ability of generative networks to convert input noise distributions into other distributions. One question they study is how easy it is to create a network that outputs *more* dimensions of noise than it receives. They derive bounds showing that an increase in dimension requires a large, complicated network. For example, an approximation of the uniform distribution on the unit square using the uniform distribution on the unit interval could use an (almost) space-filling curve such as the iterated tent map. (A space-filling curve is a continuous surjection from the unit interval to the unit square or volume of higher dimension.) This function is highly nonlinear and it can be shown that neural networks must be large to approximate it well. Thus the dimensionality of input noise is essential in practice. As we discuss later in this paper, our experiments support this conclusion for the game context.

Bailey and Telgarsky (2018) also show that, even when the input dimension is greater than the output dimension, increased input dimension can still sometimes improve accuracy. For example, one can approximate a univariate Gaussian distribution with a high dimensional uniform distribution by summing the inputs. This follows from the **Berry-Esseen theorem** (Berry, 1941), a refinement of the central limit theorem. This uses no nonlinearity, but simply takes advantage of the fact that projecting a hypercube onto a line results in an approximately Gaussian distribution.

In our experiments, we show an example of excess dimensionality improving performance.

B.3 Best response computation for continuous Colonel Blotto

Approximate best responses can be computed for the continuous Colonel Blotto game without discretizing the action space (Ganzfried, 2021). By sampling K batches of actions from other players' strategies, we can obtain an approximate best response a_i for player i using a mixed-integer linear program (MILP). More precisely, let h_{jk} be the highest bid for j from other players in batch k. Then solve the following MILP:

$$\begin{aligned}
\text{maximize} \quad & \textstyle\sum_j v_{ij} \frac{1}{K} \sum_k z_{jk} && \text{(B.13)}\\
\text{over} \quad & a_i \in \mathbb{R}^J && \text{(B.14)}\\
& z \in \{0,1\}^{J \times K} && \text{(B.15)}\\
\text{subject to} \quad & a_{ij} \geq 0 \quad \forall j && \text{(B.16)}\\
& \textstyle\sum_j a_{ij} = b_i && \text{(B.17)}\\
& z_{jk} = [a_{ij} \geq h_{jk}] \quad \forall j,k && \text{(B.18)}
\end{aligned}$$

We can use the Big M method to represent the last constraint:

$$a_{ij} - h_{jk} + M(1 - z_{jk}) \geq 0 \tag{B.19}$$

where $M \gg 0$, which forces z_{jk} to be 0 when $a_{ij} - h_{jk}$ is negative.

Appendix C

Additional experiments

In this section, we include some additional experiments.

C.1 Ablation for Fourier features in P2SNs

In this section, we study how the number of Fourier features used by a P2SN (Section 5.4) affects performance. Examples are shown in the surrounding figures. The games are as described in Section 5.4.2. The figures show that a higher number of Fourier features yields superior performance. A higher number of Fourier features allows the network to represent more levels of detail for the strategies on the player set.

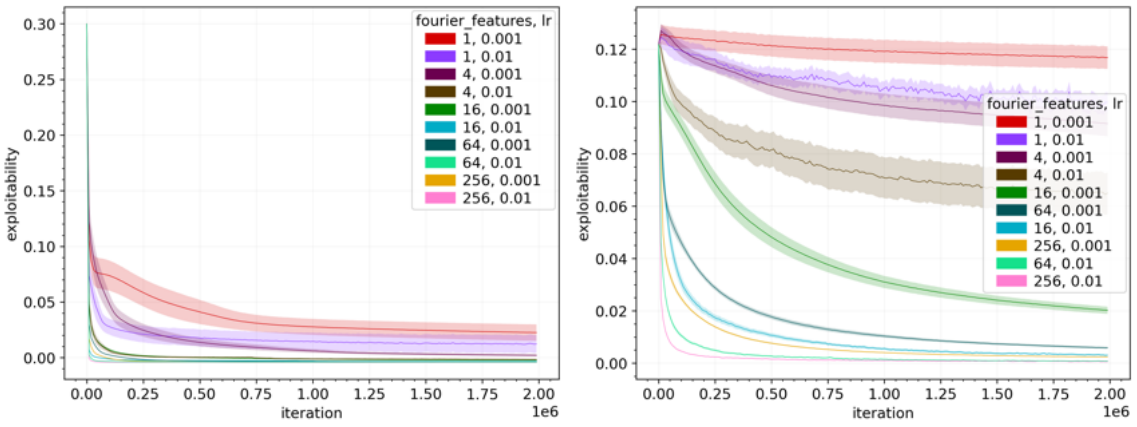


Figure C.1: Anti-coordination game. Left: 1D. Right: 2D.

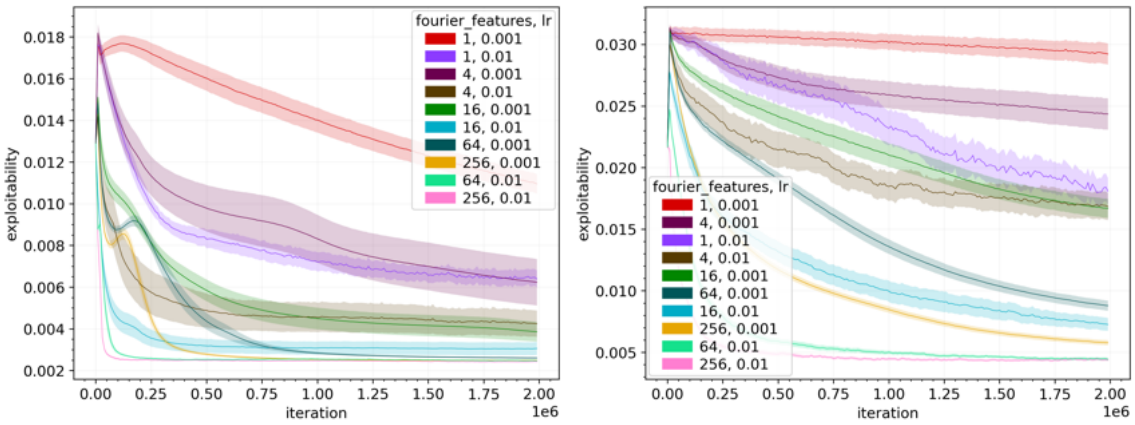


Figure C.2: Cournot game. Left: 1D. Right: 2D.

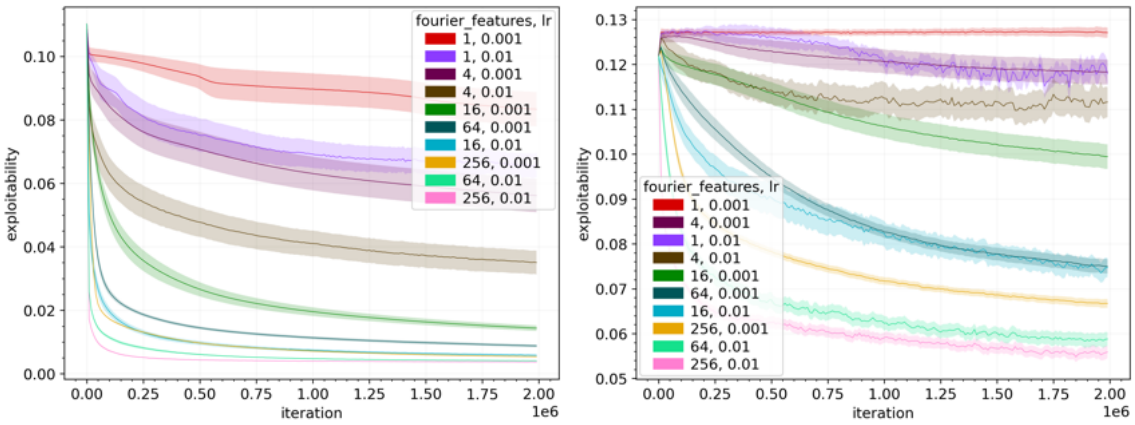


Figure C.3: Bayesian Cournot game. Left: 1D. Right: 2D.

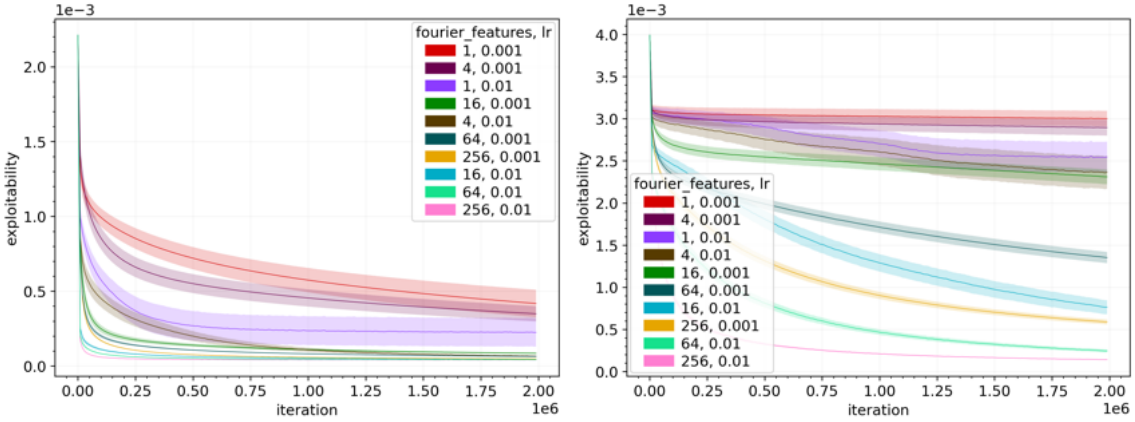


Figure C.4: Quadratic-cost Cournot game. Left: 1D. Right: 2D.

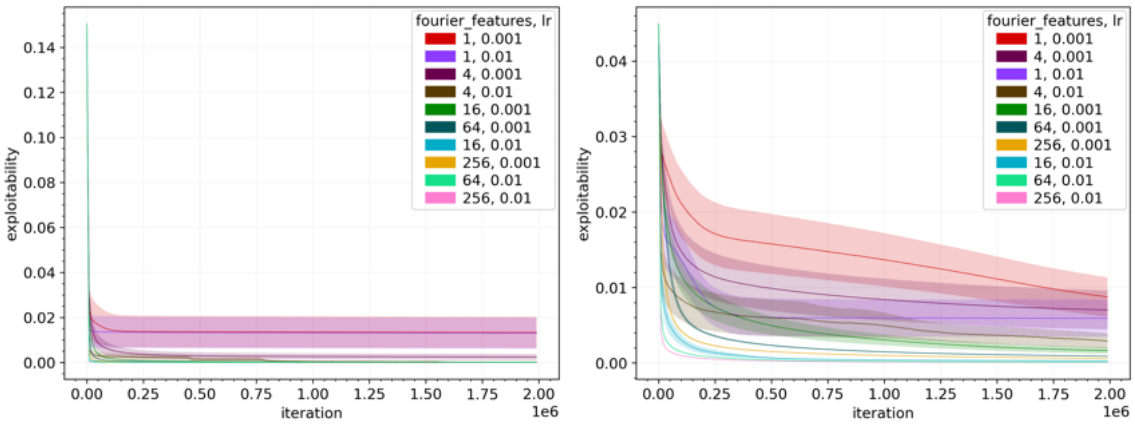


Figure C.5: Conformity game. Left: 1D. Right: 2D.

C.2 Comparison to player discretization

In this section, we compare P2SNs (Section 5.4) to an approach that discretizes the player space into a finite set of players and learns separate parameters for each. Specifically, given an infinite player space (such as the unit square), we use a low-discrepancy sequence (as described in Section 5.4.2) to obtain a finite set of points that has good coverage of the entire space. We call these points *representatives*. To select an action for a player, we "round" it to its nearest representative (in the Euclidean metric). Thus we have a Voronoi partition of the player space. Each representative/Voronoi region has its own individual parameters that are separate from the rest.

We run experiments for the anti-coordination and Cournot games described in Section 5.4.2 with this approach as well as our original approach (with a width of 96 and depth of 1). Results are shown in Figures C.6–C.13. Each plot of exploitability compares our approach (labeled "p2sn") with the discretization-based approach (labeled "discretized"). For the latter, we test various numbers of representatives (labeled "discretization_players"), thus representing different levels of resolution when discretizing the player space. We also illustrate the strategy profiles learned under the discretization approach. Visually, these approximate the strategy profiles we obtained with our approach, which were illustrated in Section 5.4.2.

We observe that, across all experiments, our approach outperforms the discretization-based approach, for every tested level of resolution. We hypothesize that the reason is the following. With a finer discretization (i.e., more representatives), it takes longer to learn because *we need to learn more independent things*. Recall that each representative has its own individual set of parameters, and is not influenced by parameter updates to other representatives. Therefore, there are "more things that need to be learned" to obtain a good strategy profile across the entire space. Conversely, with a coarser discretization (i.e., fewer representatives), there is more parameter-sharing across players overall, but it is impossible to *represent finer details* needed for good equilibria. Thus both finer and coarser discretizations suffer from their own disadvantages. In contrast, **our method gets the best of both worlds!** It is able to represent finely-detailed profiles while *simultaneously* allowing players to "learn from each other" in an indirect way, due to their shared parameters.

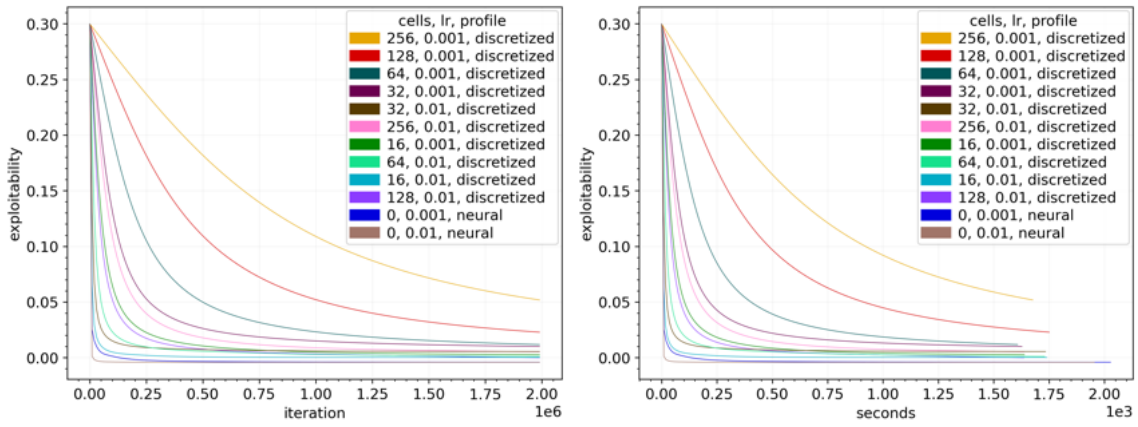


Figure C.6: 1D anti-coordination game. Exploitability. Left: Against iteration. Right: Against runtime (seconds).

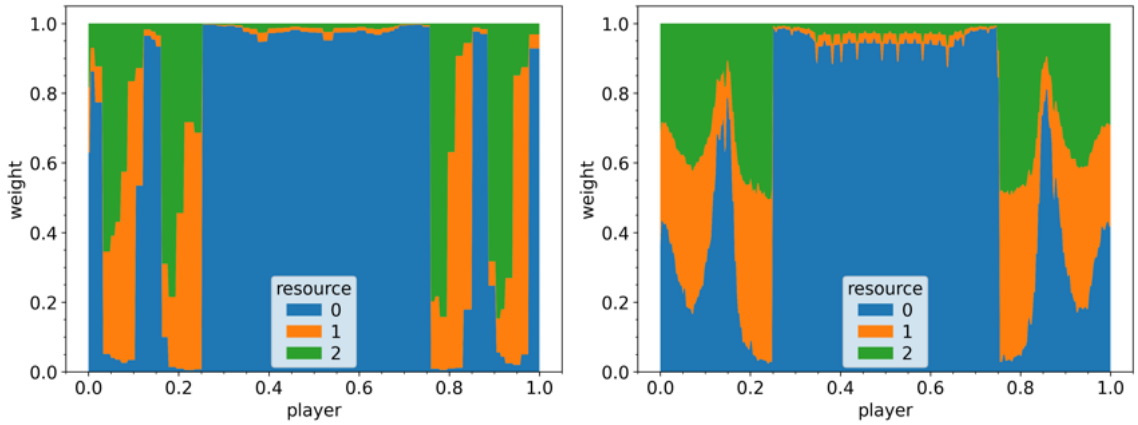


Figure C.7: 1D anti-coordination game. Learned strategy profile. Left: 64-point discretization. Right: 256-point discretization. Compare to Figure 5.39.

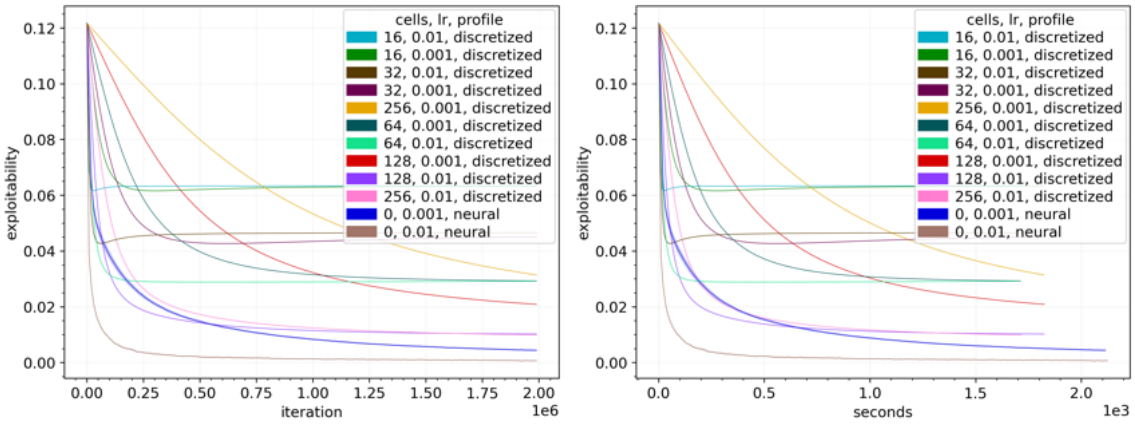


Figure C.8: 2D anti-coordination game. Exploitability. Left: Against iteration. Right: Against runtime (seconds).

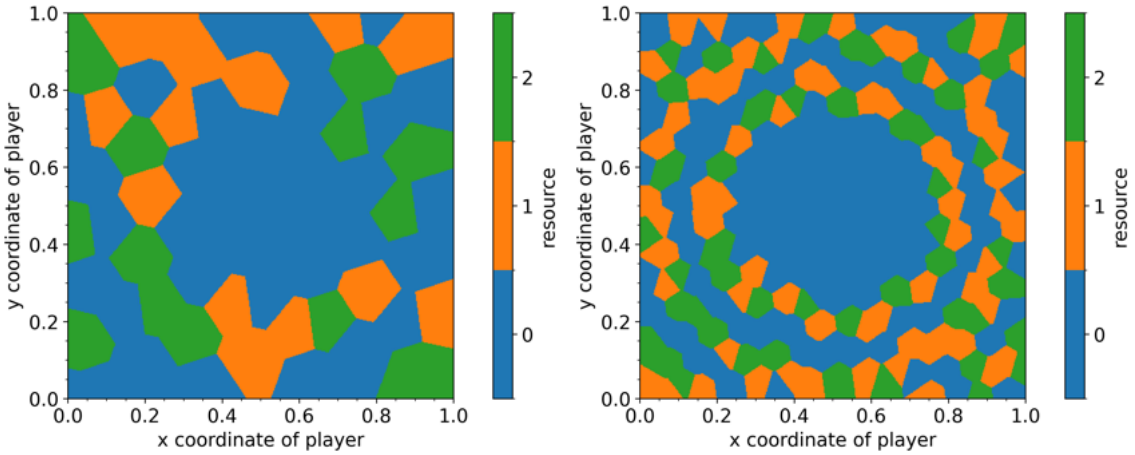


Figure C.9: 2D anti-coordination game. Learned strategy profile. Left: 64-point discretization. Right: 256-point discretization. Compare to Figure 5.40.

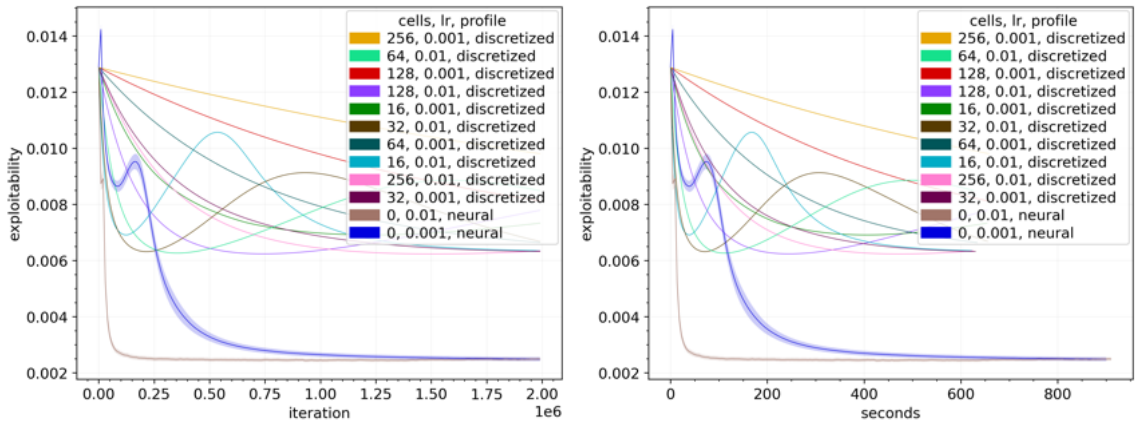


Figure C.10: 1D quadratic-cost Cournot game. Exploitability. Left: Against iteration. Right: Against runtime (seconds).

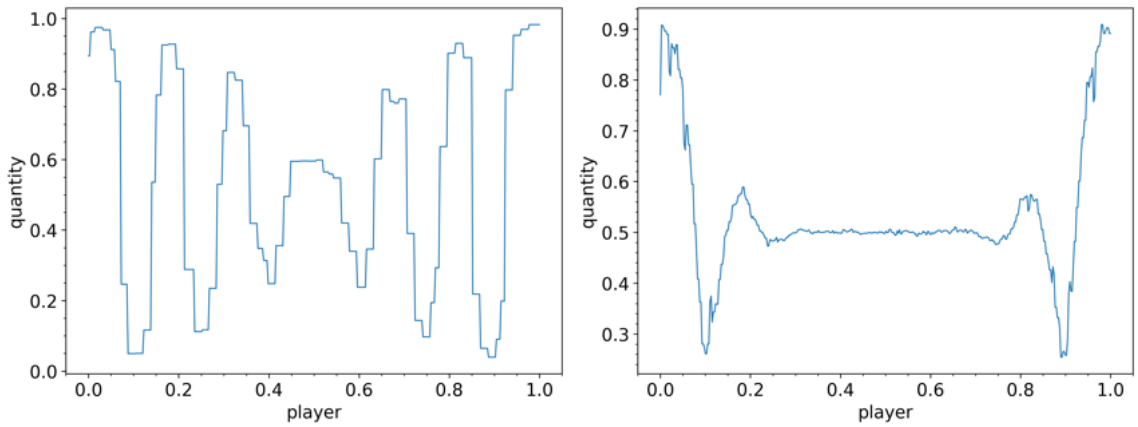


Figure C.11: 1D quadratic-cost Cournot game. Learned strategy profile. Left: 64-point discretization. Right: 256-point discretization. Compare to Figure 5.47.

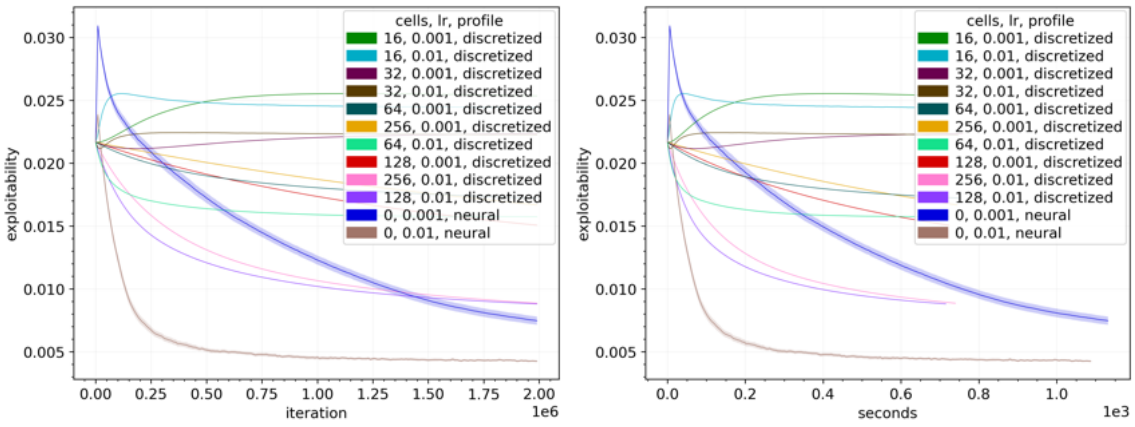


Figure C.12: 2D quadratic-cost Cournot game. Exploitability. Left: Against iteration. Right: Against runtime (seconds).

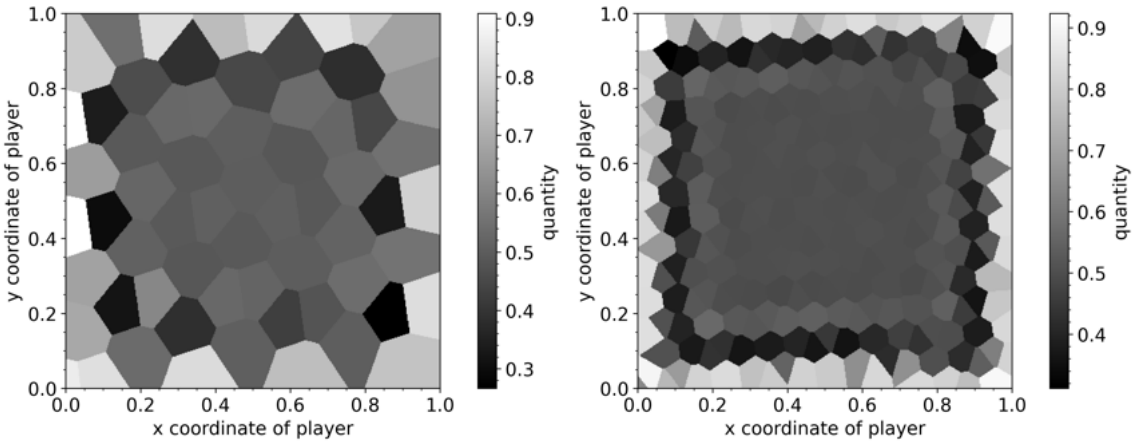


Figure C.13: 2D quadratic-cost Cournot game. Learned strategy profile. Left: 64-point discretization. Right: 256-point discretization. Compare to Figure 5.48.

Appendix D

Theorems

Here, we review some theoretical results from the literature and our completed work.

D.1 Equilibrium existence and uniqueness

Here, we review theoretical results pertaining to the existence and uniqueness of NE for various kinds of games under various conditions.

von Neumann (1928) proved the **minimax theorem**, showing that any two-player zero-sum game has an equilibrium in mixed strategies. This proof used Brouwer's fixed-point theorem (Brouwer, 1911).

Sion (1958) generalized von Neumann's minimax theorem, allowing the players' strategy sets and the utility function to have more general forms.

In *Equilibrium points in n-person games*, Nash (1950) proved that any finite game with n players has an equilibrium in mixed strategies. This proof used Kakutani's fixed-point theorem (Kakutani, 1941). In *Non-cooperative games*, Nash (1951) rewrote this proof to use Brouwer's fixed-point theorem.

D.1.1 Infinite-action games

Let (I, S, u) be an NFG, as defined in Section 3.1.

Glicksberg (1952) proved that if every S_i is nonempty and compact, and $u(s)_i$ is continuous in s, a mixed-strategy NE exists.

Fudenberg and Tirole (1991, p. 34) proved that if every S_i is nonempty, compact, and convex, and $u(s)_i$ is continuous in s and quasi-concave in s_i, a pure-strategy NE exists.

Dasgupta and Maskin (1986) proved that if every S_i is nonempty, compact, and convex, and $u(s)_i$ is upper semi-continuous and graph continuous in s and quasi-concave in s_i, a pure-strategy NE exists. They also proved that if S_i is nonempty, compact, and convex, and $u(s)_i$ is bounded and continuous in s except on a particular subset defined by technical conditions, weakly lower semi-continuous in s_i, and $\sum_{i \in I} u(s)_i$ is upper semi-continuous in s, a mixed-strategy NE exists.

Rosen (1965) proved the uniqueness of a pure-strategy NE for continuous-action games under diagonal strict concavity assumptions.

D.1.2 Infinite-player games (existence)

Schmeidler (1973) generalized Nash's theorem (Nash, 1951) on equilibrium existence to the case of a continuum of players endowed with a non-atomic measure. Khan (1985) generalized this result to non-atomic games in which each player's strategy set is a weakly compact, convex subset of a separable Banach space whose dual has the Radon–Nikodym property. Khan (1986) generalized this result to non-atomic games with strategy sets in a Banach space.

Khan and Papageorgiou (1987a) proved the existence of a Cournot–NE in generalized qualitative games with a continuum of players, under certain conditions. Similarly, Khan and Papageorgiou (1987b) proved the existence of a Cournot–NE in games with an atomless measure space of players, each with unordered preferences and strategy sets in a separable Banach space.

Kim, Prikry, and Yannelis (1989) proved the existence of an NE for an abstract economy with a measure space of agents, infinite-dimensional strategy space, and agent preferences that need not be ordered.

Rath (1992) gave a simple proof of the existence of pure-strategy NE in games with a continuum of players when a player's payoff depends only on its own action and the average action of others. This was extended to the case where the action set of each player is a compact subset of $\mathbb{R}^n$. Khan and Sun (2002) surveyed games with many players, reporting results on the existence of pure-strategy NE in games with an atomless continuum of players, each with an action set that is not necessarily finite.

Wiszniewska-Matyszkiel (2014) studied open- and closed-loop NE in games with a continuum of players, and both private and global state variables, proving an equivalence theorem between these classes of equilibria.

Yang and Song (2022) proposed an approach for proving the existence of a pure-strategy Nash equilibrium with infinitely many players as a consequence of the finite-player case.

D.1.3 Infinite-player games (uniqueness)

Milchtaich (1996) proved generic uniqueness of pure-strategy NE, and uniqueness of the equilibrium outcome, for a class of non-atomic games where a player's payoff depends on, and strictly decreases with, the measure of the set of players playing the same (pure) strategy it is playing. They also proved generic uniqueness of the Cournot–NE distribution, corresponding to a description of a game in terms of distribution of player types.

A crowding game is a game in which the payoff of each player depends only on the player's action and the size of the set of players choosing that particular action: the larger the set, the smaller the payoff. Milchtaich (2000) proved that a large crowding game generically has just one equilibrium, and the equilibrium payoffs in such a game are always unique. Moreover, the sets of equilibria of the m-replicas of a finite crowding game generically converge to a singleton as m tends to infinity.

Milchtaich (2005) proved topological conditions for uniqueness of equilibrium in physical networks with a large number of users (e.g., transportation, communication, and computer networks).

Caines, Huang, and Malhamé (2018) surveyed mean-field game theory. They presented its main results, including the existence and uniqueness of infinite-population NE, their approximating finite-population ε-NE, and the associated BR strategies.

Implications of equilibrium multiplicity for our method. As previously stated, many games have a unique NE. However, some games could have multiple NE. NE refinements and equilibrium selection are outside the scope of this project, but could be an interesting question for future research.

Even without such future study, the presented techniques can be very useful. In many-player games, game-theoretic approaches have been successful and often the most successful. For example, Pluribus (Brown and Sandholm, 2019) for multi-player no-limit Texas Hold'em is the only AI that has reached superhuman level in any large game beyond two-player zero-sum games. It is completely based on game-theoretic principles[1], and reached superhuman level without any training data. Many others had been trying to reach superhuman level on that exact problem for 70 years with rule-based approaches, supervised learning, reinforcement learning, etc. Only the game-theoretic approach succeeded. This is despite the game almost surely having a very large or infinite number of equilibria.

D.2 Exploitability

Below, convex(x) and concave(x) denote unspecified convex and concave functions of x, respectively. The calculation tracks convexity class through each operation. This is analogous to Big O notation (Bachmann, 1894; Landau, 1909; Knuth, 1976).

Definition 114. A **regular game** is an NFG (I, S, u) such that $u(s)_i$ is convex in s_{-i}.

Lemma 1. For a concave-welfare regular game, the NI is convex in the first argument.

Proof.

$$\begin{aligned}
\mathrm{NI}(x, y) &= \int_{i\sim\mu} u(x[i \mapsto y_i])_i - u(x)_i && \text{(D.1)}\\
&= \int_{i\sim\mu} u(x[i \mapsto y_i])_i - \int_{i\sim\mu} u(x)_i && \text{(D.2)}\\
&= \int_{i\sim\mu} u(x[i \mapsto y_i])_i - \mathrm{Wel}(x) && \text{(D.3)}\\
&= \int_{i\sim\mu} u(x[i \mapsto y_i])_i - \mathrm{concave}(x) && \text{(D.4)}\\
&= \int_{i\sim\mu} u(x[i \mapsto y_i])_i + \mathrm{convex}(x) && \text{(D.5)}\\
&= \left(\int_{i\sim\mu} \mathrm{convex}(x)\right) + \mathrm{convex}(x) && \text{(D.6)}\\
&= \mathrm{convex}(x) + \mathrm{convex}(x) && \text{(D.7)}\\
&= \mathrm{convex}(x) && \text{(D.8)}
\end{aligned}$$

□

Corollary 1. A concave-welfare regular game has convex exploitability.

Proof.

$$\mathrm{Expl}(x) = \sup_{y\in\Pi S} \mathrm{NI}(x, y) \tag{D.9}$$

[1] Pluribus approximated an even weaker game-theoretic solution concept than NE, namely CCE.

$$= \sup_{y \in \Pi S} \text{convex}(x) \tag{D.10}$$

$$= \text{convex}(x) \tag{D.11}$$

□

Definition 115. A **polymatrix game** is a game with a utility function of the form

$$u(x)_i = \sum_{j \neq i} x_i^\top A_{ij} x_j \tag{D.12}$$

where A_{ij} are matrices. These are graphical games in which each node corresponds to a player and each edge corresponds to a two-player bimatrix game between its endpoints. Each player chooses a single strategy for all of its bimatrix games and receives the sum of the resulting payoffs. In a *constant-sum* polymatrix game, the sum of utilities across all players is constant. In a *pairwise constant-sum* polymatrix game, the pairwise games are constant-sum.

Cai and Daskalakis (2011) motivate the class of constant-sum polymatrix games as follows:

> Intuitively, these games can be used to model a broad class of competitive environments where there is a constant amount of wealth (resources) to be split among the players of the game, with no in-flow or out-flow of wealth that may change the total sum of players' wealth in an outcome of the game
>
> As an example, they give a game that takes place in the wild west. A set of gold miners need to transport gold by splitting it into wagons that traverse different paths. Each of these paths might be controlled by thieves that could seize it.
>
> A simple example of this situation is the following game taking place in the wild west. A set of gold miners in the west coast need to transport gold to the east coast using wagons. Every miner can split her gold into a set of available wagons in whatever way she wants (or even randomize among partitions). Every wagon uses a specific path to go through the Rocky mountains.
>
> Unfortunately each of the available paths is controlled by a group of thieves. A group of thieves may control several of these paths and if they happen to wait on the path used by a particular wagon they can ambush the wagon and steal the gold being carried. On the other hand, if they wait on a particular path they will miss on the opportunity to ambush the wagons going through the other paths in their realm as all wagons will cross simultaneously.
>
> The utility of each miner in this game is the amount of her shipped gold that reaches her destination in the east coast, while the utility of each group of thieves is the total amount of gold they steal. Clearly, the total utility of all players in the wild west game is constant in every outcome of the game (it equals the total amount of gold shipped by the miners), but the pairwise interaction between every miner and group of thieves is not. In other words, the constant-sum property is a *global* rather than a *local* property of this game.

Further applications and a discussion of several special cases of these games can be found in Bergman and Fokin (1998).

Cai and Daskalakis (2011) prove a generalization of von Neumann's minmax theorem to constant-sum polymatrix games. Their theorem implies convexity of NE, polynomial-time tractability, and convergence of no-regret learning algorithms to NE.

Cai, Candogan, et al. (2016) show that, in such games, NE can be found efficiently via linear programming. They also show that the set of CCE collapses to the set of NE.

We prove that such games have convex exploitability.

Theorem 1. A constant-sum polymatrix game has convex exploitability.

Proof. For $j \neq i$, $x_i^\top A_{ij} x_j$ is linear, and therefore convex, in x_j. Therefore, their sum $u(x)_i$ is convex in x_{-i}. Furthermore, the game is constant-sum. Since the game is constant-sum, the welfare is constant, hence concave. Thus, by Corollary 1, the exploitability is convex. □

Corollary 2. A two-player constant-sum matrix game has convex exploitability.

Separately, we have the following result:

Theorem 2. A two-player constant-sum concave-convex game has convex exploitability.

Proof. The exploitability reduces to the *duality gap*, as described in Grnarova, Kilcher, et al. (2021):

$$\begin{aligned}
\operatorname{Expl}(x) &= \sup_{y_1 \in S_1} u(y_1, x_2)_1 - u(x)_1 + \sup_{y_2 \in S_2} u(x_1, y_2)_2 - u(x)_2 && \text{(D.13)}\\
&= \sup_{y_1 \in S_1} u(y_1, x_2)_1 - \inf_{y_2 \in S_2} u(x_1, y_2)_2 - u(x)_1 - u(x)_2 && \text{(D.14)}\\
&= \sup_{y_1 \in S_1} u(y_1, x_2)_1 - \inf_{y_2 \in S_2} u(x_1, y_2)_1 + \text{constant}(x) && \text{(D.15)}\\
&= \sup_{y_1 \in S_1} \text{convex}(x) - \inf_{y_2 \in S_2} \text{concave}(x) + \text{constant}(x) && \text{(D.16)}\\
&= \text{convex}(x) - \text{concave}(x) + \text{constant}(x) && \text{(D.17)}\\
&= \text{convex}(x) && \text{(D.18)}
\end{aligned}$$

□

D.3 Subgradient descent

Our techniques seek to minimize (an approximation of) exploitability by performing subgradient descent. This raises the question of when such a process is able to attain a global minimum in the first place.

Kiwiel (2004) analyze the convergence of **approximate subgradient methods** for convex optimization, and prove the following theorems. Let $\mathcal{S} \subseteq \mathbb{R}^n$ be a nonempty closed convex set, $f : \mathcal{S} \to \mathbb{R}$ be a closed proper convex function, and $\mathcal{S}_* \in \operatorname{argmin} f$. Let $x_{t+1} = P_{\mathcal{S}}(x_t - \nu_t g_t)$ where $P_{\mathcal{S}}$ is the projector onto $\mathcal{S}$ ($P_{\mathcal{S}}(x) \in \operatorname{argmin}_{y \in \mathcal{S}} \|x - y\|$), $\nu_t \geq 0$ is a stepsize, $\varepsilon_t \geq 0$ is an error tolerance, and $g_t \in \partial_{\varepsilon_t} f(x_t)$ is an ε_t-*approximate subgradient* of f at x_t, that is, $f(x) \geq f(x_t) + \langle g_t, x - x_t \rangle - \varepsilon_t$ for all x.

Theorem 3. (Kiwiel, 2004, Theorem 3.4) Suppose $\mathcal{S}_* \neq \varnothing$, $\sum_{t \in \mathbb{N}} \nu_t = \infty$, and $\sum_{t \in \mathbb{N}} \nu_t (\frac{1}{2} \|g_t\|^2 \nu_t + \varepsilon_t) < \infty$. Then $\{x_t\}_{t \in \mathbb{N}}$ converges to some $x_\infty \in \mathcal{S}_*$.

Theorem 4. (Kiwiel, 2004, Theorem 3.6) Suppose $\mathcal{S}_* \neq \varnothing$, $\sum_{t\in\mathbb{N}} \nu_t = \infty$, $\sum_{t\in\mathbb{N}} \nu_t^2 < \infty$, $\sum_{t\in\mathbb{N}} \nu_t \varepsilon_t < \infty$, and the subgradients do not grow too fast: $\exists c < \infty, \forall t \in \mathbb{N}, \|g_t\|^2 \leq c(1 + \|x_t\|^2)$ (e.g., they are bounded). Then $\{x_t\}_{t\in\mathbb{N}}$ converges to some $x_\infty \in \mathcal{S}_*$.

Convergence results are also known for subgradient methods on *quasi*-convex functions, which is a more general class of functions. Some of these are described in Hu, Yang, and Sim (2015).

In our paper, we seek to approximately minimize the exploitability function $\Phi \in \mathcal{S} \to \mathbb{R}$, $\Phi(x) = \sup_{y\in\mathcal{S}} \phi(x, y)$, where ϕ is the Nikaido–Isoda function and $\mathcal{S}$ is the set of possible strategy profiles. Specifically, we use subgradients of $\tilde{\Phi}_t(x) = \phi(x, \tilde{y}_t)$, where $\tilde{y}_t$ is the response profile output by the learned **best-response ensembles** (BRE) or **best-response function** (BRF) at t. Thus $\tilde{\Phi}_t(x) \geq \tilde{\Phi}_t(x_t) + \langle g_t, x - x_t\rangle$. Unconditionally, $\Phi \geq \tilde{\Phi}_t$ (since the former maximizes over all possible y). Thus $\Phi(x) \geq \tilde{\Phi}_t(x_t) + \langle g_t, x - x_t\rangle$.

Now, suppose we can guarantee that $\tilde{\Phi}(x_t) \geq \Phi(x_t) - \varepsilon_t$ for an error tolerance $\varepsilon_t \geq 0$; that is, the responses output by the BRE/BRF at t do not perform *too* badly (in the limit) compared to the true best responses. Then $\Phi(x) \geq \Phi(x_t) - \varepsilon_t + \langle g_t, x - x_t\rangle$.

Therefore, when the assumptions of the above theorems hold, the sequence of iterates $\{x_t\}_{t\in\mathbb{N}}$ converges to a global minimizer of the exploitability function, which is an NE, if an NE exists at all.

D.4 Theoretical results pertaining to P2SN

In this section, we include some theoretical analysis of the method we proposed in Martin and Sandholm (2025e).

D.4.1 Progress guarantee

Let

$$g = \left\{ \frac{\mathrm{d}u(s_\theta)_i}{\mathrm{d}(s_\theta)_i} \right\}_{i\in P} \tag{D.19}$$

be the true simultaneous gradient in the (potentially infinite-dimensional) **function space** of strategy profiles. Let

$$v = \left[\int_{i\sim\mu} \frac{\mathrm{d}}{\mathrm{d}\theta} u(s_\phi[i \mapsto (s_\theta)_i])_i \right]_{\phi=\theta} \tag{D.20}$$

$$= \int_{i\sim\mu} \frac{\mathrm{d}(s_\theta)_i}{\mathrm{d}\theta} \frac{\mathrm{d}u(s_\theta)_i}{\mathrm{d}(s_\theta)_i} \tag{D.21}$$

be the SPSG. The change in strategy profile induced by the SPSG is

$$\tilde{g} = Jv \tag{D.22}$$

where $J = \frac{\mathrm{d}s_\theta}{\mathrm{d}\theta}$ is the network Jacobian. Furthermore,

$$v = \int_{i\sim\mu} J_i g_i \tag{D.23}$$

$$= J^\top g \tag{D.24}$$

Therefore,

$$\begin{aligned} g \cdot \tilde{g} &= g \cdot (Jv) && \text{Eq. D.22} && \text{(D.25)} \\ &= (J^\top g) \cdot v && && \text{(D.26)} \\ &= v \cdot v && \text{Eq. D.24} && \text{(D.27)} \\ &= \|v\|^2 && && \text{(D.28)} \\ &\geq 0 && && \text{(D.29)} \end{aligned}$$

That is, the true simultaneous gradient and SPSG-induced gradient are compatible. Furthermore,

$$v \neq 0 \leftrightarrow g \notin \text{kernel}(J^\top) \leftrightarrow g \not\perp \text{im}(J) \tag{D.30}$$

Therefore,

$$g \cdot \tilde{g} > 0 \leftrightarrow g \notin \text{kernel}(J^\top) \leftrightarrow g \not\perp \text{im}(J) \tag{D.31}$$

That is, if the simultaneous gradient has a nonzero projection onto the tangent space of the network—i.e., onto the Jacobian—progress is made. Any components of the true simultaneous gradient that are orthogonal to the tangent space cannot be represented by any parameter update. If $g \neq 0$ but $v = 0$, then the network-representable tangent space cannot implement any average improvement direction. In practice, this means that the network must be expressive enough to move in directions that improve players' utilities on average.

D.4.2 Special cases

In this section, we develop stronger conclusions under various additional assumptions about the setting.

Disjoint parameters. Suppose $(s_\theta)_i$ depends only on θ_i. Then

$$\begin{aligned} v(\theta) &= \int_{i \sim \mu} \left. \frac{\mathrm{d}u(s_\phi[i \mapsto (s_\theta)_i])_i}{\mathrm{d}\theta} \right|_{\phi=\theta} && \text{(D.32)} \\ &= \int_{i \sim \mu} \left. \frac{\mathrm{d}u(s_\phi[i \mapsto (s_\theta)_i])_i}{\mathrm{d}\theta_i} \right|_{\phi=\theta} \frac{\mathrm{d}\theta_i}{\mathrm{d}\theta} && \text{(D.33)} \\ &= \int_{i \sim \mu} \frac{\mathrm{d}u(s_\theta)_i}{\mathrm{d}\theta_i} \mathbf{e}_i && \text{(D.34)} \end{aligned}$$

Thus SPSG reduces to the usual simultaneous gradient.

Potential game. Suppose the game is a potential game. Let ϕ be its potential function. Then

$$u(s[i \mapsto a])_i - u(s[i \mapsto b])_i = \phi(s[i \mapsto a]) - \phi(s[i \mapsto b]) \tag{D.35}$$

Thus

$$\frac{\mathrm{d}u(s)_i}{\mathrm{d}s_i} = \frac{\mathrm{d}\phi(s)}{\mathrm{d}s_i} \tag{D.36}$$

Let L be the potential composed with the strategy network.

$$L(\theta) = \phi(s_\theta) \tag{D.37}$$

On the one hand,

$$\begin{aligned}
\frac{\mathrm{d}L(\theta)}{\mathrm{d}\theta} &= \frac{\mathrm{d}\phi(s_\theta)}{\mathrm{d}\theta} && \text{(D.38)}\\
&= \int_{i\sim\mu} \frac{\mathrm{d}\phi(s_\theta)}{\mathrm{d}(s_\theta)_i}\frac{\mathrm{d}(s_\theta)_i}{\mathrm{d}\theta} && \text{(D.39)}\\
&= \int_{i\sim\mu} \frac{\mathrm{d}u(s_\theta)_i}{\mathrm{d}(s_\theta)_i}\frac{\mathrm{d}(s_\theta)_i}{\mathrm{d}\theta} && \text{Eq. D.36} \quad \text{(D.40)}
\end{aligned}$$

On the other hand,

$$\begin{aligned}
v(\theta) &= \int_{i\sim\mu} \left.\frac{\mathrm{d}u(s_\phi[i\mapsto(s_\theta)_i])_i}{\mathrm{d}\theta}\right|_{\phi=\theta} && \text{(D.41)}\\
&= \int_{i\sim\mu} \left.\frac{\mathrm{d}u(s_\phi[i\mapsto(s_\theta)_i])_i}{\mathrm{d}(s_\theta)_i}\right|_{\phi=\theta}\frac{\mathrm{d}(s_\theta)_i}{\mathrm{d}\theta} && \text{(D.42)}\\
&= \int_{i\sim\mu} \frac{\mathrm{d}u(s_\theta)_i}{\mathrm{d}(s_\theta)_i}\frac{\mathrm{d}(s_\theta)_i}{\mathrm{d}\theta} && \text{(D.43)}
\end{aligned}$$

Therefore

$$v(\theta) = \frac{\mathrm{d}L(\theta)}{\mathrm{d}\theta} \tag{D.44}$$

That is, SPSG is equivalent to stochastic gradient ascent on the network-transformed potential. For example, if ϕ is convex and $\theta \mapsto s_\theta$ is affine in θ, L is also convex. All theoretical guarantees for stochastic gradient ascent carry over to this setting, given appropriate bounded-variance and integrability conditions on the integrand of $v(\theta)$.

Concave utilities. Suppose that each player's utility is concave and differentiable in its own strategy. Let $\ell = -u$ be the loss function. (Here, we work with losses because it is more common in the optimization literature.) Then each player's loss is convex and differentiable in its own strategy. Suppose that $(s_\theta)_i = A_i\theta + b_i$ is affine. Thus $\nabla^2_{s_i}\ell(s)_i \succeq 0$ for all s.

Let $\tilde{\ell}(\theta) = \ell(r[i\mapsto(s_\theta)_i])_i$, where r is the strategy profile that player i is deviating from. Then

$$\begin{aligned}
\nabla_\theta\tilde{\ell}(\theta) &= (\nabla_\theta(s_\theta)_i)(\nabla_{(s_\theta)_i}\tilde{\ell}(\theta)) && \text{(D.45)}\\
&= A_i^\top(\nabla_{(s_\theta)_i}\tilde{\ell}(\theta)) && \text{(D.46)}
\end{aligned}$$

and

$$\begin{aligned}
\nabla^2_\theta\tilde{\ell}(\theta) &= A_i^\top(\nabla^2_{(s_\theta)_i}\tilde{\ell}(\theta))A_i && \text{(D.47)}\\
&= A_i^\top(\nabla^2_{(s_\theta)_i}\ell(s_\theta))A_i && \text{(D.48)}
\end{aligned}$$

By hypothesis $\nabla^2_{s_i}\ell(s)_i \succeq 0$, thus $A_i^\top(\nabla^2_{s_i}\ell(s)_i)A_i \succeq 0$. Therefore, the symmetric part of $\nabla^2_\theta\tilde{l}(\theta)$ is PSD. Integrating over i preserves this, so the symmetric part of $\nabla_\theta v(\theta)$ is PSD. Since $v(\theta)$ is continuously differentiable and its symmetric Jacobian is PSD, it is a monotone operator. Thus, the theoretical convergence guarantees for stochastic monotone operators apply. Loizou et al. (2021) gives an example of a general convergence guarantee for such operators.