Combining Deep Reinforcement Learning and Search with Generative Models for Game-Theoretic Opponent Modeling

Zun Li[1] , **Marc Lanctot**[1] , **Kevin R. McKee**[1] , **Luke Marris**[1] , **Ian Gemp**[1] , **Daniel Hennes**[1] , **Paul Muller**[1] , **Kate Larson**[1,2] , **Yoram Bachrach**[1] , **Michael P. Wellman**[3]
[1]Google DeepMind
[2]University of Waterloo
[3]University of Michigan
{lizun,lanctot}@google.com

Abstract

Opponent modeling methods typically involve two crucial steps: building a belief distribution over opponents' strategies, and exploiting this opponent model by playing a best response. However, existing approaches typically require domain-specific heurstics to come up with such a model, and algorithms for approximating best responses are hard to scale in large, imperfect information domains.

In this work, we introduce a scalable and generic multiagent training regime for opponent modeling using deep game-theoretic reinforcement learning. We first propose Generative Best Respoonse (GenBR), a best response algorithm based on Monte-Carlo Tree Search (MCTS) with a learned deep generative model that samples world states during planning. This new method scales to large imperfect information domains and can be plug and play in a variety of multiagent algorithms. We use this new method under the framework of Policy Space Response Oracles (PSRO), to automate the generation of an *offline opponent model* via iterative game-theoretic reasoning and population-based training. We propose using solution concepts based on bargaining theory to build up an opponent mixture, which we find identifying profiles that are near the Pareto frontier. Then GenBR keeps updating an *online opponent model* and reacts against it during gameplay. We conduct behavioral studies where human participants negotiate with our agents in Deal-or-No-Deal, a class of bilateral bargaining games. Search with generative modeling finds stronger policies during both training time and test time, enables online Bayesian co-player prediction, and can produce agents that achieve comparable social welfare and Nash bargaining score negotiating with humans as humans trading among themselves.

1 Introduction

A central challenge for agent designers is how to build agents that can be well-adapted to unknown opponents in a dynamic multiagent environment. Opponent modeling methods [Albrecht and Stone, 2018] typically build a profile or a prior belief of opponent strategies and produce agents that are best response against such opponent models. These techniques have achieved success in fields like Poker [Johanson and Bowling, 2009; Bard *et al.*, 2013], automated negotiation [Baarslag *et al.*, 2016] and robotic soccer [Kitano *et al.*, 1998].

However, most of these approaches use domain-specific heurstics to handcraft an opponent model. Such knowledge is typically encoded in certain interpretations of game rules or experiences reflected by human plays. These techniques are hard to transfer to domains where relevant data are missing. Meanwhile, even if an opponent model is present, there is no existing best response method that work well in large imperfect information games where computing a posterior distribution over world state is intractable.

In this work, we propose a general-purpose training regime using multiagent reinforcement learning to address the above issues. We adopt extensive-form games (EFG) as a generic formulation for multiagent environments for our algorithmic developments, as opposed to a domain-specific ruleset. We propose Generative Best Response (GenBR), a best response method that extends AlphaZero-style RL and MCTS methods to large general-sum, imperfect information games. GenBR enhances best response strength by leveraging test-time computation and an approximate world model learned by deep neural nets. While GenBR can be plug and play in a variety of multiagent training algorithms, we focus on *Policy Space Response Oracles* (PSRO) [Lanctot *et al.*, 2017] as our training loop for *offline opponent modeling*. Our agent thereafter at test-time employs GenBR for both planning and updating an *online opponent model* via Bayesian learning.

Contributions. We provide three significant contributions. First, we propose enhanced version of AlphaZero-style MCTS to train a best response strategy, thereby equipping our agent with the capability to both plan and infer the environmental state as well as opponents' strategic choices during online decision-making. This novel search method integrates deep RL with *Information Set MCTS* (IS-MCTS). To handle large imperfect information, we augment the policy-and-value network (PVN) in AlphaZero with a generative model that samples world states at the root of the search tree. This results in a novel policy-value-and-generative network (PVGN), which iteratively refines its quality using RL trajec-

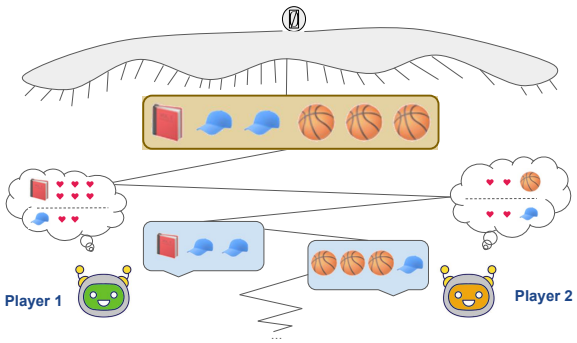

Figure 1: Example negotiation game in extensive-form. In "Deal or No Deal", the game starts at the empty history ($\varnothing$), chance samples a public pool of resources and private preferences (with some conditions; see Section 5.1) for three different items (books, hats, and basketballs). Then, players alternate proposals for how to split the resources. Rewards are defined by a dot product between received resources and preferences.

tory data during the training loop.

Second, we introduce several novel meta-strategy solvers in the PSRO framework based on the Nash bargaining solution [Nash, 1950]. Lastly, we show an empirical evaluation of several RL agents trained by the combined algorithm against humans in a negotiation game, finding the best ones to be as efficient as humans trading with other humans.

2 Preliminaries

An N-player ***normal-form game*** consists of a set of players $\mathcal{N} = \{1, 2, \ldots, N\}$, N finite pure strategy sets Π_i (one per player) with a joint strategy set $\Pi = \Pi_1 \times \Pi_2 \times \cdots \Pi_N$, and a utility tensor (one per player), $u_i : \Pi \to \mathbb{R}$, and we denote player i's utility as $u_i(\pi)$. Two-player (2P) normal-form games are called ***matrix games***. A ***two-player zero-sum*** (purely adversarial) game is such that, $N = 2$ and for all joint strategies $\pi \in \Pi : \sum_{i \in \mathcal{N}} u_i(\pi) = 0$, whereas a ***common-payoff*** (purely cooperative) game: $\forall \pi \in \Pi, \forall i, j \in \mathcal{N} : u_i(\pi) = u_j(\pi)$. A ***general-sum*** game is one without any restrictions on the utilities. A ***mixed strategy*** for player i is a probability distribution over Π_i denoted $\sigma_i \in \Delta(\Pi_i)$, and a strategy profile $\sigma = \sigma_1 \times \cdots \times \sigma_N$, and for convenience we denote $u_i(\sigma) = \mathbb{E}_{\pi \sim \sigma}[u_i(\pi)]$. By convention, $-i$ refers to player i's opponents. A ***best response*** is a strategy $b_i(\sigma_{-i}) \in \text{BR}(\sigma_{-i}) \subseteq \Delta(\Pi_i)$, that maximizes the utility against a specific opponent strategy: for example, $\sigma_1 = b_1(\sigma_{-1})$ is a best response to σ_{-1} if $u_1(\sigma_1, \sigma_{-1}) = \max_{\sigma_1'} u_1(\sigma_1', \sigma_{-1})$. An approximate ***$\epsilon$-Nash equilibrium*** is a profile σ such that for all $i \in \mathcal{N}, u_i(b_i(\sigma_{-i}), \sigma_{-i}) - u_i(\sigma) \leqslant \epsilon$, with $\epsilon = 0$ corresponding to an exact Nash equilibrium.

In an ***extensive-form game***, play takes place over a sequence of actions $a \in \mathcal{A}$. Examples of such games include chess, Go, and poker. A ***history*** $h \in \mathcal{H}$ is a sequence of actions from the start of the game taken by all players. Legal actions are at h are denoted $\mathcal{A}(h)$ and the player to act at h as $\tau(h)$. Players only partially observe the state and hence have imperfect information. There is a special player called ***chance*** that plays with a fixed stochastic policy (selecting outcomes that represent dice rolls or private preferences). Policies π_i (also called behavioral strategies) is a collection of distributions over legal actions, one for each player's ***information state***, $s \in \mathcal{S}_i$, which is a set of histories consistent with what the player knows at decision point s (*e.g.*all the possible private preferences of other players), and $\pi_i(s) \in \Delta(\mathcal{A}(s))$. An illustrative example of interaction in an extensive-form game is "Deal or No Deal" shown in Figure 1.

There is a subset of the histories $\mathcal{Z} \subset \mathcal{H}$ called ***terminal histories***, and utilities are defined over terminal histories, e.g. $u_i(z)$ for $z \in \mathcal{Z}$ could be –1 or 1 in Go (representing a loss and a win for player i, respectively). As before, expected utilities of a joint profile $\pi = \pi_1 \times \cdots \times \pi_N$ is defined as an expectation over the terminal histories, $u_i(\pi) = \mathbb{E}_{z \sim \pi}[u_i(z)]$, and best response and Nash equilibria are defined with respect to a player's full policy space.

2.1 Combining MCTS and RL for Best Response

Computing approximate best responses is critical in a variety of multiagent training algorithms. There have been many previous works that consider enhancing best response strength via deep RL and game-tree search. For example, the original PSRO algorithm [Lanctot *et al.*, 2017] proposed using deep RL in place of value iteration in the double-oracle framework [McMahan *et al.*, 2003] which scales game solving in environments with large state spaces. Approximate Best Response (ABR) [Timbers *et al.*, 2022; Wang *et al.*, 2023] uses MCTS as a best response method which identifies weakness in human-level Go playing agents. However, they only focus on domain where a posterior world belief can be computed exactly. Another relevant work is Best Response Expert Iteration (BRExIT) [Hernandez *et al.*, 2023], which uses MCTS to exploit an opponent mixture trained with an auxiliary loss. However, they only concern perfect information games.

In general, using deep RL+MCTS as a best response method leverages both generalization capability of neural nets and test-time computation by the search method. We elaborate this by explaining how ABR algorithm works. When computing an approximate best response in imperfect information games, ABR uses a variant of Information Set Monte Carlo tree search [Cowling *et al.*, 2012] called IS-MCTS-BR. At the root of the IS-MCTS-BR search (starting at information set s), the posterior distribution over world states, $\Pr(h \mid s, \pi_{-i})$ is computed explicitly, which requires both (i) enumerating every history in s, and (ii) computing the opponents' reach probabilities for each history in s. Then, during each search round, a world state is sampled from this belief distribution, then the game-tree regions are explored in a similar way as in the vanilla MCTS, and finally the statistics are aggregated on the information-set level. Steps (i) and (ii) are prohibitively expensive in games with large belief spaces.

Algorithm 1 GenBR Training Loop

function GenBR(i, σ, num_eps)
  Initialize value nets $\boldsymbol{v}, \boldsymbol{v}'$, policy nets $\boldsymbol{p}, \boldsymbol{p}'$, generative nets $\boldsymbol{g}, \boldsymbol{g}'$, data buffers $D_{\boldsymbol{v}}, D_{\boldsymbol{p}}, D_{\boldsymbol{g}}$
  for $eps = 1, \ldots, num\_eps$ **do**
    $h \leftarrow$ initial state. $\mathcal{T} = \{s_i(h)\}$
    Sample opponents $\pi_{-i} \sim \sigma_{-i}$.
    while h not terminal **do**
      if $\tau(h) = chance$ **then**
        Sample chance event $a \sim \pi_c$
      else if $\tau(h) \neq i$ **then**
        Sample $a \sim \pi_{\tau(h)}$
      else
        $a, \pi \leftarrow$ Search$(s_i(h), \sigma, \boldsymbol{v}', \boldsymbol{p}', \boldsymbol{g}')$
        $D_{\boldsymbol{p}} \leftarrow D_{\boldsymbol{p}} \bigcup \{(s_i(h), \pi)\}$
        $D_{\boldsymbol{g}} \leftarrow D_{\boldsymbol{g}} \bigcup \{(s_i(h), h)\}$
      end if
      $h \leftarrow h.apply(a)$, $\mathcal{T} \leftarrow \mathcal{T} \bigcup \{s_i(h)\}$
    end while
    $D_{\boldsymbol{v}} \leftarrow D_{\boldsymbol{v}} \bigcup \{(s, r) \mid s \in \mathcal{T}\}$, where r is the payoff of i in this trajectory
    $\boldsymbol{v}, \boldsymbol{p}, \boldsymbol{g} \leftarrow$ Update$(\boldsymbol{v}, \boldsymbol{p}, \boldsymbol{g}, D_{\boldsymbol{v}}, D_{\boldsymbol{p}}, D_{\boldsymbol{g}})$
    Replace parameters of $\boldsymbol{v}', \boldsymbol{p}', \boldsymbol{g}'$ by the latest parameters of $\boldsymbol{v}, \boldsymbol{p}, \boldsymbol{g}$ periodically.
  end for
  return Search$(\cdot, \sigma, \boldsymbol{v}, \boldsymbol{p}, \boldsymbol{g})$
end function

Algorithm 2 GenBR Search

function Search($s, \sigma, \boldsymbol{v}, \boldsymbol{p}, \boldsymbol{g}$)
  for $iter = 1, \ldots, num\_sim$ **do**
    $\mathcal{T} = \{\}$
    Sample a world state (gen. model): $h \sim \boldsymbol{g}(h \mid s)$
    Sample an opponent profile using Bayes' rule: $\pi'_{-i} \sim \Pr(\pi_{-i} \mid h, \sigma_{-i})$. Replace opponent nodes with chance events according to π'_{-i}
    while do
      if h is terminal **then**
        $r \leftarrow$ payoff of i. **Break**
      else if $\tau(h) = chance$ **then**
        $a \leftarrow$ sample according to chance
      else if $s_i(h)$ not in search tree **then**
        Add $s_i(h)$ to search tree.
        $r \leftarrow \boldsymbol{v}(s_i(h))$
      else
        $a \leftarrow$ MaxPUCT$(s_i(h), \boldsymbol{p})$
        $\mathcal{T} \leftarrow \mathcal{T} \cup \{(s_i(h), a)\}$
      end if
      $h.apply(a)$
    end while
    for $(s, a) \in \mathcal{T}$ **do**
      $s.child(a).visits \leftarrow s.child(a).visits + 1$
      $s.child(a).value \leftarrow s.child(a).value + r$
      $s.total\_visits \leftarrow s.total\_visits + 1$
    end for
  end for
  return action a^* that receives max visits among children of s, and a policy π^* that represents the visit frequency of children of s
end function

3 GenBR: Learning Best Response Search with a Generative Model

We propose Generative Best Response (GenBR), a new best response method based on AlphaZero-styled MCTS with $\Pr(h \mid s, \pi_{-i})$ learned by a deep generative model. On a high-level, GenBR is parameterized by three deep neural nets: a policy net $\boldsymbol{p}$, a value net $\boldsymbol{v}$ and a generative net $\boldsymbol{g}$. Just as in the AlphaZero training loop [Silver *et al.*, 2018], GenBR training loop generates multiple RL trajectories by calling GenBR search procedure at each decision point to gather data for training these neural nets. These neural nets will further guide and refine the search procedure, producing higher quality data for later training. During the search procedure, the world states are sampled directly from the model given only their information state descriptions, leading to a succinct representation of the posterior that generalizes to large state spaces. Next we explain our algorithms in more details.

The GenBR training loop (Algorithm 1) proceeds analogously to AlphaZero's self-play based training, which trains a value net $\boldsymbol{v}$, a policy net $\boldsymbol{p}$, along with a generative network $\boldsymbol{g}$ using trajectories generated by search. We assume we are given an *offline opponent model* σ represented by a mixed strategy profile of the opponents. There are important differences from AlphaZero. Only one player is learning (*e.g.*player i). The (set of) opponents are fixed, sampled at the start of each episode from the opponent model σ_{-i}. Whenever it is player i's turn to play, since we are considering imperfect information games, it runs a POMDP search procedure based on Algorithm 2 from its current information state s_i. The search procedure produces a policy target π^*, and an action choice a^* that will be taken at s_i at that episode. Data about the final outcome and policy targets for player i are stored in data sets $D_{\boldsymbol{v}}$ and $D_{\boldsymbol{p}}$, which are used to improve the value net and policy net that guide the search. Data about the history, h, in each information set, $s(h)$, reached is stored in a data set $D_{\boldsymbol{g}}$, which is used to train the generative network $\boldsymbol{g}$ by supervised learning. We call the combination of $\boldsymbol{v}$, $\boldsymbol{p}$, and $\boldsymbol{g}$ as the policy-value-and-generative network (PVGN).

The GenBR MCTS search used (Algorithm 2) is based on IS-MCTS-BR in [Timbers *et al.*, 2022] (described in Section 2.1) and POMCP [Silver and Veness, 2010]. Here it utilizes value net $\boldsymbol{v}$ to truncate the search at an unexpanded node and policy net $\boldsymbol{p}$ for action selection at an expanded node s using the PUCT [Silver *et al.*, 2018] formula: $\text{MaxPUCT}(s, \boldsymbol{p}) = \arg\max_{a \in \mathcal{A}(s)} \frac{s.child(a).value}{s.child(a).visits} + c_{uct} \cdot \boldsymbol{p}(s, a) \cdot \frac{\sqrt{s.total\_visits}}{s.child(a).visits+1}$, for some constant c_{uct}. Then at the end of the search call, it returns an action a^* which receives the most visits at the root node, and a policy π^* representing the action distribution of the search at the root node.

Algorithm 2 has two important differences from previous methods. First, rather than computing exact posteriors, we use the deep generative model $\boldsymbol{g}$ learned in Algorithm 1 to sample world states. As such, this approach may be capable of scaling to large domains where previous approaches such as particle filtering [Somani *et al.*, 2013] fail.

Second, the imperfect information of the underlying POMDP consists of both (i) the actual world state h and (ii)

opponents' pure-strategy commitment π_{-i}. We make use of the fact $\Pr(h, \pi_{-i} \mid s, \sigma_{-i}) = \Pr(h \mid s, \sigma_{-i}) \Pr(\pi_{-i} \mid h, \sigma_{-i})$ such that we approximate $\Pr(h \mid s, \sigma_{-i})$ by $\boldsymbol{g}$ and compute $\Pr(\pi_{-i} \mid h, \sigma_{-i})$ exactly via Bayes' rule. Computing $\Pr(\pi_{-i} \mid h, \sigma_{-i})$ can be viewed as a Bayesian learning procedure [Kreps and Wilson, 1982; Hernandez-Leal and Kaisers, 2017; Albrecht *et al.*, 2016; Kalai and Lehrer, 1993] that keeps updating an *online opponent model* $\Pr(\pi_{-i} \mid h, \sigma_{-i})$. Therefore, our agent is capable of performing test-time search while automatically inferring environmental state as well as opponents' strategies during online decision making.

Algorithm 3 Opponent Modeling via PSRO and GenBR

function PSRO($\mathcal{G}$, MSS)
  Initialize strategy sets $\forall i, \Pi_i = \{\pi_i^0\}$, mixed strategies $\sigma_i(\pi_i^0) = 1, \forall i$, payoff tensor U^0.
  for $t \in \{0, 1, 2 \cdots, T\}$ **do**
    for $i \in \mathcal{N}$ **do**
      $\Pi_i \leftarrow \Pi_i \bigcup \{\texttt{GenBR}(i, \sigma, num\_eps)\}$
    end for
    Update missing entries in U^t via simulations
    $\sigma \leftarrow \text{MSS}(U^t)$
  end for
  return $\Pi = (\Pi_1, \Pi_2, \cdots, \Pi_N), \sigma$
end function

4 Game-Theoretic Opponent Modeling

In this section, we describe our complete training algorithm for opponent modeling. We use Policy Space Response Oracles to obtain an *offline opponent model* σ_{-i} for training GenBR in Algorithm 1. We first introduce empirical game-theoretic analysis (EGTA) and PSRO, and then propose new solution concepts in PSRO based on bargaining theory, and provide empirical results of our new meta-strategy solvers at the end of this section.

4.1 Empirical Game-Theoretic Analysis and Policy-Space Response Oracles

Empirical game-theoretic analysis (EGTA) [Wellman *et al.*, 2025] is an approach to reasoning about large sequential games through normal-form ***empirical game*** models, induced by simulating enumerated subsets of the players' full policies in the sequential game. Policy-Space Response Oracles (PSRO) [Lanctot *et al.*, 2017] uses EGTA to incrementally build up each player's set of policies ("oracles") through repeated applications of approximate best response using RL. Each player's initial set contains a single policy (*e.g.*uniform random) resulting in a trivial empirical game U^0 containing one cell. On epoch t, given N sets of policies Π_i^t for $i \in \mathcal{N}$, utility tensors for the empirical game U^t are estimated via simulation.

A ***meta-strategy solver*** (MSS) derives a profile σ^t, generally mixed, over the empirical game strategy space. A best response oracle, say $b_i^t(\sigma_{-i}^t)$, is then computed for each player i by training against policies sampled from opponent model σ_{-i}^t, and are added to strategy sets for the next epoch: $\Pi_i^{t+1} = \Pi_i^t \cup \{b_i^t(\sigma_{-i}^t)\}$. Since the opponent policies are fixed, the oracle response step is a single-agent problem [Oliehoek and Amato, 2014]; RL and search can feasibly handle large state and policy spaces. Our full algorithm (Algorithm 3) employs GenBR as the oracle step in PSRO.

PSRO naturally fits our focus of automating opponent modeling as it generates opponent models σ_{-i} through pure game-theoretic reasoning and reinforcement learning. Furthermore, since each strategy (except the first ones) in the pool is a best response against an early-iteration opponent model (which itself consists of best responses), it in fact induces a cognitive hierarchy [Camerer *et al.*, 2004] of rationalizable strategies [Bernheim, 1984]. An important question is choosing which MSS[1] to compute an opponent mixture. Since our primary application domain of interest in this paper is negotiation game, it is natural to consider solution concepts from bargaining theory as MSSs. Next we introduce computational results of new MSSs based on Nash bargaining solution which were not investigated in previous works.

4.2 Empirical Game Nash Bargaining Solution

In contrast to non-cooperative game theory, bargaining theory considers scenarios where players' utilities are not entirely in conflict, and need to negotiate to achieve a possible cooperative outcome. The ***Nash Bargaining solution*** (NBS) selects a Pareto-optimal payoff profile that uniquely satisfies axioms specifying desirable properties of invariance, symmetry, and independence of irrelevant alternatives [Nash, 1950; Ponsati and Watson, 1997]. The axiomatic characterization of NBS abstracts away the process by which said outcomes are obtained through strategic interaction.

Define the set of achievable payoffs as all expected utilities $u_i(\mathbf{x})$ under a joint-policy profile $\mathbf{x}$. Denote the disagreement outcome of player i, which is the payoff it gets if no agreement is achieved, as d_i. The NBS is the set of policies that maximizes the ***Nash bargaining score*** (A.K.A. *Nash product*):

$$\max_{\mathbf{x} \in \Delta(\Pi)} \Pi_{i \in \mathcal{N}} \left(u_i(\mathbf{x}) - d_i\right), \tag{1}$$

which, when $N = 2$, leads to a quadratic program (QP) with the constraints derived from the policy space structure [Griffin, 2010]. However, even in this simplest case of two-player matrix games, the non-concave objective poses a problem for most QP solvers. Furthermore, scaling to N players requires higher-order polynomial solvers.

Instead of using higher-order polynomial solvers, we propose an algorithm based on (projected) gradient ascent [Singh *et al.*, 2000; Boyd and Vandenberghe, 2004]. Note that the Nash product is non-concave, so instead of maximizing it, we maximize the ***log Nash product*** $g(\mathbf{x}) =$

$$\log\left(\Pi_{i \in \mathcal{N}}(u_i(\mathbf{x}) - d_i)\right) = \sum_{i \in \mathcal{N}} \log(u_i(\mathbf{x}) - d_i), \tag{2}$$

which has the same maximizers as (1), and is a sum of concave functions, hence concave. The process is depicted in

[1] In App. G.1, we conduct extensive experiments over 16 MSSs on 12 different benchmark games on OpenSpiel [Lanctot *et al.*, 2019]. Appendices can be found in [Li *et al.*, 2023].

Algorithm 4 NBS by projected gradient ascent

Input: Initial iterate $\mathbf{x}$, payoff tensor U.
function $\texttt{NBS}(\mathbf{x}^0, U)$
  Let $g(\mathbf{x})$ be the log Nash product defined in eqn (2)
  for $t = 0, 1, 2 \cdots, T$ **do**
    $\mathbf{y}^{t+1} \leftarrow \mathbf{x}^t + \alpha^t \nabla g(\mathbf{x}^t)$
    $\mathbf{x}^{t+1} \leftarrow \texttt{Proj}(\mathbf{y}^{t+1})$
  end for
  Return $\arg\max_{\mathbf{x}^{t=0:T}} g(\mathbf{x}^t)$
end function

Algorithm 4; $\texttt{Proj}$ is the ℓ_2 projection onto the simplex. We can prove it has a convergence rate of $O(T^{-1/2})$. For a proof, see App. C.

Theorem 4.1. *Assume any deal is better than no deal by $\kappa > 0$, i.e., $u_i(\mathbf{x}) - d_i \geqslant \kappa > 0$ for all $i, \mathbf{x}$. Let $\{\mathbf{x}^t\}$ be the sequence generated by Algorithm 4 with starting point $\mathbf{x}^0 = |\Pi|^{-1}\mathbf{1}$ and step size sequence $\alpha^t = \frac{\kappa\sqrt{(|\Pi|-1)/|\Pi|}}{u^{\max}N}(t+1)^{-1/2}$. Then, for all $t > 0$ one has*

$$\max_{\mathbf{x}\in\Delta^{|\Pi|-1}} g(\mathbf{x}) - \max_{0\leqslant s\leqslant t} g(\mathbf{x}^s) \leqslant \frac{u^{\max}N\sqrt{|\Pi|}}{\kappa\sqrt{t+1}} \tag{3}$$

where $u^{\max} = \max_{i,\mathbf{x}} u_i(\mathbf{x})$, $|\Pi|$ is the number of possible pure joint strategies, and $\mathbf{x}$ is assumed to be a joint correlation device (μ).

Besides directly using NBS, we also consider (1) using NBS to select a correlated equilibrium (CE) and coarse correlated equilibrium (CCE) as an MSS, which we denote as max-NBS-(C)CE and (2) profile that maximizes social welfare. A comprehensive list of MSSs that we investigated in our experiments is in App. A.

4.3 Empirical Results on Colored Trails

Here we study the performance of our NBS-based MSSs on colored trails, a highly configurable negotiation game played on a grid [Gal *et al.*, 2010] of colored tiles, which has been actively studied by the AI community [Grosz *et al.*, 2004; Ficici *et al.*, 2008]. Colored Trails does not require search since the number of moves is small, so we use classical RL based oracles (DQN and Boltzmann DQN) to isolate the effects of the new meta-strategy solvers.

We use a three-player variant [Ficici *et al.*, 2008] depicted in Figure 2. At the start of each episode, an instance (a board and colored chip allocation per player) is randomly sampled from a database of strategically interesting and balanced configurations [de Jong *et al.*, 2011, Section 5.1]. There are two proposers (P1 and P2) and a responder (R). R can see all players' chips, both P1 and P2 can see R's chips; however, proposers cannot see each other's chips. Each proposer, makes an offer to the receiver. The receiver than decides to accept one offer and trades chips with that player, or passes. Then, players spend chips to get as close to the flag as possible (each chip allows a player to move to an adjacent tile if it is the same color as the chip). For any configuration (player i at position p), define $\textsc{Score}(p, i) = (-25)d + 10t$, where d is the Manhattan distance between p and the flag, and t is the number of player i's chips. The utility for player i is their *gain*: score at the end of the game minus the score at the start. We compute the Pareto frontier for a subset of configurations, and define the Pareto Gap (P-Gap) as the minimal ℓ_2 distance from the outcomes to the outer surfaces of the convex hull of the Pareto front, which is then averaged over the set of configurations in the database.

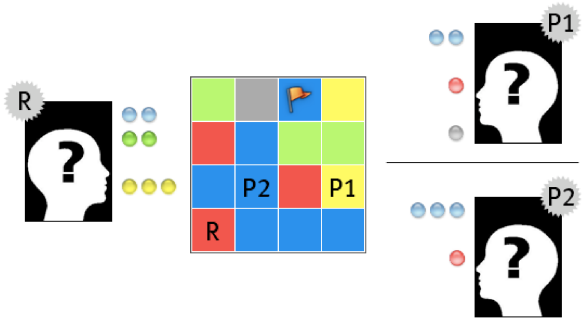

Figure 2: Three-Player Colored Trails.

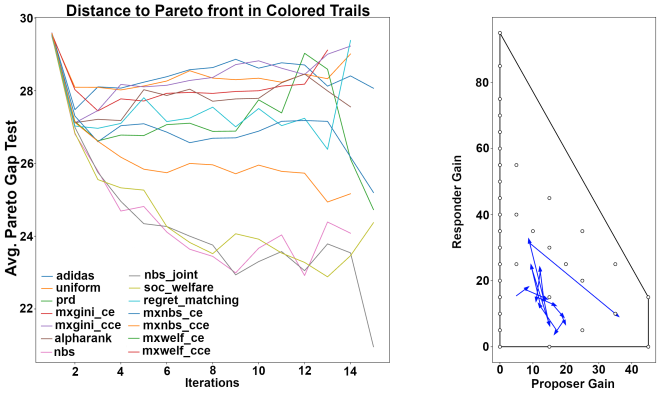

Figure 3: Empirical reduction in Pareto Gap on test game configurations, and example evolution toward Pareto front (right).

Figure 3 shows representative results of PSRO agents with different MSSs on Colored Trails (for full graphs, and evolution of score diagrams, see App. G.2). The best-performing MSS is NBS-joint, beating the next best by a full 3 points. The NBS meta-strategy solvers comprise five of the six best MSSs under this evaluation. An example of the evolution of the expected score over PSRO iterations is also shown, moving toward the Pareto front, though not via a direct path.

5 Experiments

In this section, we report our major results on the Deal-or-No-Deal (DoND) negotiation game.

5.1 Negotiation Game: Deal or No Deal

"Deal or No Deal" (DoND) is a simple alternating-offer bargaining game with incomplete information, which has been used in many AI studies [Lewis *et al.*, 2017; Cao *et al.*, 2018]. Our focus is to train RL agents to play against humans *without human data*, similar to previous work [Strouse *et al.*, 2021].

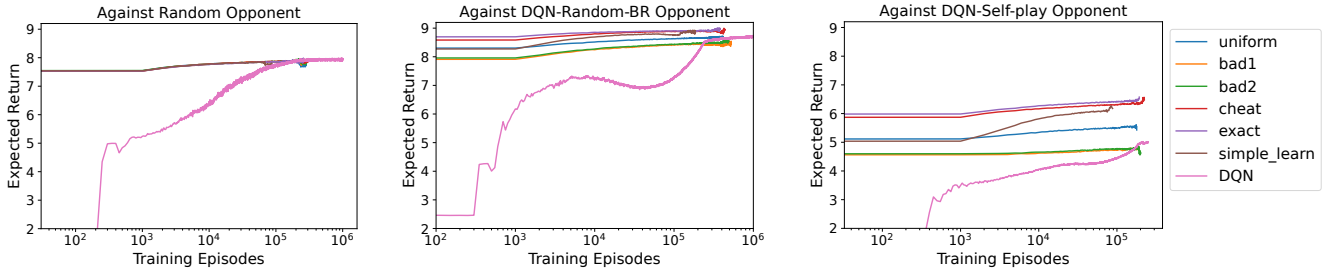

Figure 4: Best response performance using different generative models, against (left) uniform random opponent, (middle) DQN response to uniform random, (right) self-play DQN opponent. **Uniform** samples a legal preference vector uniformly at random, **bad1** always samples the first legal instance in the database, **bad2** always samples the last legal instance in the database, **cheat** always samples the actual underlying world state, **exact** samples from the exact posterior, **simple learn** is the method described in Algorithm 1, and **DQN** is a simple DQN responder that does not use a generative model nor search. All results are averaged across 30 random seeds.

An example game of DoND is shown in Figure 1. Two players are assigned *private* preferences $\boldsymbol{w}_1 \geqslant \mathbf{0}, \boldsymbol{w}_2 \geqslant \mathbf{0}$ for three different items (books, hats, and basketballs). At the start of the game, there is a pool $\boldsymbol{c}$ of 5 to 7 items drawn randomly such that: (i) the total value for a player of all items is 10: $\boldsymbol{w}_1 \cdot \boldsymbol{c} = \boldsymbol{w}_2 \cdot \boldsymbol{c} = 10$, (ii) each item has non-zero value for at least one player: $\boldsymbol{w}_1 + \boldsymbol{w}_2 > \mathbf{0}$, (iii) some items have non-zero value for both players, $\boldsymbol{w}_1 \odot \boldsymbol{w}_2 \neq \mathbf{0}$, where $\odot$ represents element-wise multiplication.

The players take turns proposing how to split the pool of items, for up to 10 turns (5 turns each). If an agreement is not reached, the negotiation ends and players both receive 0. Otherwise, the agreement represents a split of the items to each player, $\boldsymbol{o}_1 + \boldsymbol{o}_2 = \boldsymbol{c}$, and player i receives a utility of $\boldsymbol{w}_i \cdot \boldsymbol{o}_i$. DoND is an imperfect information game because the other player's preferences are private. We use a database of 6796 bargaining instances made publicly available in [Lewis *et al.*, 2017]. Deal or No Deal is a significantly large game, with an estimated $1.32 \cdot 10^{13}$ information states for player 1 and $5.69 \cdot 10^{11}$ information states for player 2 (see App. D.2 for details).

5.2 Generative World State Sampling

We now show that both the search and the generative model contribute to achieving higher reward than RL alone. The input of our deep generative model is one's private values $\mathbf{v}_i$ and public observations, and the output is a distribution over $\mathbf{v}_{-i}$ (detailed in the Appendix). We compute approximate best responses to three opponents: uniform random, a DQN agent trained against uniform random, and a DQN agent trained in self-play. We compare different world state sampling models as well to DQN in Figure 4, where the deep generative model approach is denoted as simple_learn.

The benefit of search is clear: the search methods achieve a high value in a few episodes, a level that takes DQN many more episodes to reach (against random and DQN response to random) and a value that is not reached by DQN against the self-play opponent. The best generative models are the true posterior (exact) and the actual underlying world state (cheat). However, the exact posterior is generally intractable and the underlying world state is not accessible to the agent at test-time, so these serve as idealistic upper-bounds. Uniform seems to be a compromise between the bad and ideal models. The deep generative model approach is roughly comparable to uniform at first, but learns to approximate the posterior as well as the ideal models as data is collected. In contrast, DQN eventually reaches the performance of the uniform model against the weaker opponent but not against the stronger opponent even after 20000 episodes.

5.3 Studies with Human Participants

We recruited participants from Prolific [Pe'er *et al.*, 2021; Pe'er *et al.*, 2017] to evaluate the performance of our agents in DoND (overall 346; 41.4% female, 56.9% male, 0.9% trans or nonbinary; median age range: 30–40), following established ethical guidelines for research with human participants [McKee, 2024]. Crucially, participants played DoND for real monetary stakes, with an additional payout for each point earned in the game. Participants first read game instructions and completed a short comprehension test to ensure they understood key aspects of DoND's rules. Participants then played five episodes of DoND with a randomized sequence of opponents. Episodes terminated after players reached a deal, after 10 moves without reaching a deal, or after 120 seconds elapsed. After playing all five episodes, participants completed a debrief questionnaire collecting standard demographic information and open-ended feedback on the study.

Training Details Our infrastructure restricts that each human participant can only play five matches with our bots. Therefore we decided to select five different agents so every participant can play each of these once. For comparison, we decided to include one independent RL agent and four search-improved PSRO agents of different playing styles. For the independent RL agent, we trained two classes of independent RL agents in self-play: (1) DQN [Mnih *et al.*, 2015] and Boltzmann DQN [Cui and Koeppl, 2021], and (2) policy gradient algorithms such as A2C, QPG, RPG and RMPG [Srinivasan *et al.*, 2018]. We fine-tuned their hyperparameters and eventually select an instance of self-play DQN based on both individual returns and social welfare performances.

For PSRO agents, we consider 16 different meta-strategy solvers, and 4 different back-propagating value types during the tree search procedure, making it 64 different combina-

Agent	$\bar{u}_{\text{Humans}}$		$\bar{u}_{\text{Agent}}$		$\bar{u}_{\text{Comb}}$		NBS
IndRL	5.86	[5.37, 6.40]	**6.50**	**[5.93, 7.06]**	6.18	[5.82, 6.56]	38.12
Comp1	5.14	[4.56, 5.63]	5.49	[4.87, 6.11]	5.30	[4.93, 5.76]	28.10
Comp2	6.00	[5.49, 6.55]	5.54	[4.96, 6.10]	5.76	[5.33, 6.12]	33.13
Coop	6.71	[6.23, 7.20]	6.17	[5.66, 6.64]	6.44	[6.11, 6.75]	41.35
Fair	**7.39**	**[6.89, 7.87]**	5.98	[5.44, 6.49]	**6.69**	**[6.34, 7.01]**	**44.23**

Table 1: Humans vs. agents performance with **129** human participants, **547** games total. $\bar{u}_{\boldsymbol{X}}$ refers to the average utility to group $\boldsymbol{X}$ (for the humans when playing the agent, or for the agent when playing the humans), Comb refers to Combined (human and agent). Square brackets indicate 95% confidence intervals. IndRL refers to Independent RL (DQN), Comp1 and Comp2 are the two top-performing competitive agents, Coop is the most cooperative agent, and Fair is fairest agent. NBS is the Nash bargaining score (Eq 1).

tion in total. Notice that the original MCTS algorithm (Algorithm 2) back-propagates individual rewards during each simulation phase for the search agent.

We consider two way of extracting the final agents given sets of policies Π produced by PSRO, one based on the final-iteration MSS mixture and another based on the final-iteration search-based best response (details are in App F). We trained over 100 PSRO agents with different combinations of MSS, back-propagation targets, and extraction methods. To select among these agents, we apply empirical game-theoretic analysis [Wellman *et al.*, 2025] and obtain a head-to-head empirical game matrix. We eventually selected: (i) two most competitive agents (Comp1, Comp2) (maximizing utility), (ii) the most cooperative agents (Coop) (maximizing social welfare), the (iii) the fairest agent (Fair) (minimizing social inequity [Fehr and Schmidt, 1999]); (iv) a separate top-performing independent RL agent (IndRL) trained in self-play (DQN). Both Coop and Fair are using Nash product as the back-propagating values during tree search, while Comp1 uses inequity aversion and Comp2 uses individual rewards. Comp1, Comp2 and Fair are trained using max-Gini correlated equilibrium notions [Marris *et al.*, 2021], while Coop uses uniform distribution as the MSS.

Results We collect data under two conditions: human vs. human (HvH), and human vs. agent (HvA). In the HvH condition, we collect 483 games: 482 end in deals made (99.8%), and achieve a return of 6.93 (95% c.i. [6.72, 7.14]), on expectation. We collect 547 games in the HvA condition: 526 end in deals made (96.2%; see Table 1). DQN achieves the highest individual return. By looking at the combined reward, it achieves this by aggressively reducing the human reward (down to 5.86)–possibly by playing a policy that is less human-compatible. The competitive PSRO agents seem to do the same, but without overly exploiting the humans, resulting in the lowest social welfare overall. The cooperative agent achieves significantly higher combined utility playing with humans. Better yet is Human vs. Fair, the only Human vs. Agent combination to achieve social welfare comparable to the Human vs. Human social welfare.

Another metric is the objective value of the empirical NBS from Eq. 1, over the symmetric game (randomizing the starting player) played between the different agent types. This metric favors Pareto-efficient outcomes, balancing between the improvement obtained by both sides. From App G.3, the NBS of Coop decreases when playing humans, from $44.51 \rightarrow 41.35$– perhaps due to overfitting to agent-learned conventions. Fair increases slightly ($42.56 \rightarrow 44.23$). The NBS of DQN rises from $23.48 \rightarrow 38.12$. The NBS of the competitive agents also rises playing against humans ($24.70 \rightarrow 28.10$, and $25.44 \rightarrow 28.10$), and also when playing with Fair ($24.70 \rightarrow 29.63$, $25.44 \rightarrow 28.73$). The fair agent is both adaptive to many different types of agents, and cooperative, increasing the social welfare in all groups it negotiated with. This could be due to its MSS putting significant weight on many policies leading to Bayesian prior with high support, or its backpropagation of the product of utilities rather than individual return.

6 Conclusion

We proposed a general-purpose multiagent training regime that combines the power of MCTS search and a population-based training framework, for general-sum imperfect information domains. We developed a novel search technique that combines IS-MCTS with a deep belief learning module coupled with the RL training loop, which scale to large belief and state spaces. The outer loop of our algorithm is implemented by PSRO, which iteratively trains and adds search strategies guided by game-theoretic analysis. On one hand, search serves as a strong best response method within the PSRO loop, which provides an instance of the framework of its own interests. On the other hand, PSRO automatically produces a belief hierarchy over the opponents' strategies, which endows the search with the capability of inferring opponent types during online decision makings. This dual view of the whole training architecture illustrates its effectiveness in producing agents that are capable of opponent modeling through game-theoretic analysis and planning forward at test-time.

Ethics Statement

We believe our GenBR method with the PSRO training loop advances the general opponent modeling and planning techniques in multi-agent systems with little domain knowledge. Our methods can be potentially deployed in a variety of applications, including automated bidding in auctions, negotiation, cybersecurity, warehouse robotics, and autonomous vehicle systems. All of these are multi-agent scenarios that involve general-sum, imperfect information elements.

One of the potential risks is value misalignment in negotiation. The method can produce strategies that are unpredictable and not easily explained, which could lead to exploitative behaviors in negotiation that are misaligned with

the users' intents. This could potentially cause harm in the economic system and reduce market efficiency. Any deployed use of artificial agents built using our algorithm would need to first be thoroughly tested, ideally by third party, and undergo a controlled private study with humans to identify any potentially harmful behavior.

References

[Albrecht and Stone, 2018] Stefano V Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

[Albrecht *et al.*, 2016] Stefano V. Albrecht, Jacob W. Crandall, and Subramanian Ramamoorthy. Belief and truth in hypothesised behaviours. *Artificial Intelligence*, 235:63–94, 2016.

[Baarslag *et al.*, 2016] Tim Baarslag, Mark JC Hendrikx, Koen V. Hindriks, and Catholijn M. Jonker. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems*, 30(5):849–898, 2016.

[Bard *et al.*, 2013] Nolan Bard, Michael Johanson, Neil Burch, and Michael Bowling. Online implicit agent modelling. In *Proceedings of the Twelfth International Conference on Autonomous Agents and Multi-Agent Systems*, 2013.

[Bernheim, 1984] B. Douglas Bernheim. Rationalizable strategic behavior. *Econometrica*, 52:1007–1028, 1984.

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[Camerer *et al.*, 2004] Colin F. Camerer, Teck-Hua Ho, and Juin-Kuan Chong. A cognitive hierarchy model of games. *Quarterly Journal of Economics*, 119(3):861–898, 2004.

[Cao *et al.*, 2018] Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z. Leibo, Karl Tuyls, and Stephen Clark. Emergent communication through negotiation. In *Sixth International Conference on Learning Representations*, 2018.

[Cowling *et al.*, 2012] Peter I. Cowling, Edward J. Powley, and Daniel Whitehouse. Information set Monte Carlo tree search. *IEEE Transactions on Computational Intelligence and AI in Games*, 4:120–143, 2012.

[Cui and Koeppl, 2021] Kai Cui and Heinz Koeppl. Approximately solving mean field games via entropy-regularized deep reinforcement learning. In *Twenty-Fourth International Conference on Artificial Intelligence and Statistics*, 2021.

[de Jong *et al.*, 2011] Steven de Jong, Daniel Hennes, Karl Tuyls, and Ya'akov Gal. Metastrategies in the colored trails game. In *Tenth International Conference on Autonomous Agents and Multi-Agent Systems*, pages 551–558, 2011.

[Fehr and Schmidt, 1999] E. Fehr and K. Schmidt. A theory of fairness, competition and cooperation. *Quarterly Journal of Economics*, 114:817–868, 1999.

[Ficici *et al.*, 2008] Sevan G. Ficici, David C. Parkes, and Avi Pfeffer. Learning and solving many-player games through a cluster-based representation. In *Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 188–195, 2008.

[Gal *et al.*, 2010] Ya'akov Gal, Barbara Grosz, Sarit Kraus, Avi Pfeffer, and Stuart Shieber. Agent decision-making in open mixed networks. *Artificial Intelligence*, 174(18):1460–1480, 2010.

[Griffin, 2010] Christopher Griffin. Quadratic programs and general-sum games. In *Game Theory: Penn State Math 486 Lecture Notes*, pages 138–144. Online note., 2010. https://docs.ufpr.br/~volmir/Math486.pdf.

[Grosz *et al.*, 2004] Barbara J. Grosz, Sarit Kraus, Shavit Talman, Boaz Stossel, and Moti Havlin. The influence of social dependencies on decision-making: Initial investigations with a new game. In *Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 782–789, 2004.

[Hernandez *et al.*, 2023] Daniel Hernandez, Hendrik Baier, and Michael Kaisers. Brexit: On opponent modelling in expert iteration. In *Thirty-Second International Joint Conference on Artificial Intelligence*, 2023.

[Hernandez-Leal and Kaisers, 2017] Pablo Hernandez-Leal and Michael Kaisers. Learning against sequential opponents in repeated stochastic games. In *Third Multi-disciplinary Conference on Reinforcement Learning and Decision Making*, volume 25, 2017.

[Johanson and Bowling, 2009] Michael Johanson and Michael Bowling. Data biased robust counter strategies. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 264–271, 2009.

[Kalai and Lehrer, 1993] Ehud Kalai and Ehud Lehrer. Rational learning leads to Nash equilibrium. *Econometrica: Journal of the Econometric Society*, pages 1019–1045, 1993.

[Kitano *et al.*, 1998] Hiroaki Kitano, Milind Tambe, Peter Stone, Manuela Veloso, Silvia Coradeschi, Eiichi Osawa, Hitoshi Matsubara, Itsuki Noda, and Minoru Asada. The robocup synthetic agent challenge 97. In *RoboCup-97: Robot Soccer World Cup I 1*, pages 62–73, 1998.

[Kreps and Wilson, 1982] David M. Kreps and Robert Wilson. Reputation and imperfect information. *Journal of Economic Theory*, 27(2):253–279, 1982.

[Lanctot *et al.*, 2017] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Thirty-First International Conference on Neural Information Processing Systems*, pages 4190–4203, 2017.

[Lanctot *et al.*, 2019] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. Openspiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*, 2019.

[Lewis *et al.*, 2017] Mike Lewis, Denis Yarats, Yann N. Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning for negotiation dialogues. In *Conference on Empirical Methods in Natural Language Processing*, 2017.

[Li *et al.*, 2023] Zun Li, Marc Lanctot, Kevin R McKee, Luke Marris, Ian Gemp, Daniel Hennes, Paul Muller, Kate Larson, Yoram Bachrach, and Michael P Wellman. Combining deep reinforcement learning and search with generative models for game-theoretic opponent modeling. *arXiv preprint arXiv:2302.00797*, 2023.

[Marris *et al.*, 2021] Luke Marris, Paul Muller, Marc Lanctot, Karl Tuyls, and Thore Graepel. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. In *Twenty-Eighth International Conference on Machine Learning*, 2021.

[McKee, 2024] Kevin R McKee. Human participants in AI research: Ethics and transparency in practice. *IEEE Transactions on Technology and Society*, 2024.

[McMahan *et al.*, 2003] H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *Twentieth International Conference on International Conference on Machine Learning*, page 536–543, 2003.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Nash, 1950] John Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[Oliehoek and Amato, 2014] Frans A. Oliehoek and Christopher Amato. Best response Bayesian reinforcement learning for multiagent systems with state uncertainty. In *Ninth AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains*, 2014.

[Pe'er *et al.*, 2017] Eyal Pe'er, Laura Brandimarte, Sonam Samat, and Alessandro Acquisti. Beyond the Turk: Alternative platforms for crowdsourcing behavioral research. *Journal of Experimental Social Psychology*, 70:153–163, 2017.

[Pe'er *et al.*, 2021] Eyal Pe'er, David Rothschild, Andrew Gordon, Zak Evernden, and Ekaterina Damer. Data quality of platforms and panels for online behavioral research. *Behavior Research Methods*, pages 1–20, 2021.

[Ponsati and Watson, 1997] Clara Ponsati and Joel Watson. Multiple-issue bargaining and axiomatic solutions. *International Journal of Game Theory*, 62:501–524, 1997.

[Silver and Veness, 2010] David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Twenty-Fourth International Conference on Neural Information Processing Systems*, volume 23, 2010.

[Silver *et al.*, 2018] David Silver, 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. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[Singh *et al.*, 2000] Satinder Singh, Michael Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000.

[Somani *et al.*, 2013] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. In *Twenty-Seventh International Conference on Neural Information Processing Systems*, volume 26, 2013.

[Srinivasan *et al.*, 2018] Sriram Srinivasan, Marc Lanctot, Vinicius Zambaldi, Julien Pérolat, Karl Tuyls, Rémi Munos, and Michael Bowling. Actor-critic policy optimization in partially observable multiagent environments. In *Thirty-First International Conference on Neural Information Processing Systems*, 2018.

[Strouse *et al.*, 2021] DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In *Thirty-Fifth International Conference on Neural Information Processing Systems*, pages 14502–14515, 2021.

[Timbers *et al.*, 2022] Finbarr Timbers, Nolan Bard, Edward Lockhart, Marc Lanctot, Martin Schmid, Neil Burch, Julian Schrittwieser, Thomas Hubert, and Michael Bowling. Approximate exploitability: Learning a best response in large games. In *Thirty-First International Joint Conference on Artificial Intelligence*, pages 3487–3493, 2022.

[Wang *et al.*, 2023] Tony Tong Wang, Adam Gleave, Nora Belrose, Tom Tseng, Joseph Miller, Kellin Pelrine, Michael D Dennis, Yawen Duan, Viktor Pogrebniak, Sergey Levine, and Stuart Russell. Adversarial policies beat superhuman Go AIs. In *Fortieth International Conference on Machine Learning*, 2023.

[Wellman *et al.*, 2025] Michael P Wellman, Karl Tuyls, and Amy Greenwald. Empirical game theoretic analysis: A survey. *Journal of Artificial Intelligence Research*, 82:1017–1076, 2025.