

# Sample-Efficient Policy Space Response Oracles with Joint Experience Best Response

Ariyan Bighashdel  
Utrecht University  
Utrecht, The Netherlands  
Delft University of Technology  
Delft, The Netherlands  
a.bighashdel@uu.nl

Thiago D. Simão  
Eindhoven University of Technology  
Eindhoven, The Netherlands  
t.simao@tue.nl

Frans A. Oliehoek  
Delft University of Technology  
Delft, The Netherlands  
f.a.oliehoek@tudelft.nl

## ABSTRACT

Multi-agent reinforcement learning (MRL) offers a scalable alternative to exact game-theoretic analysis but suffers from non-stationarity and the need to maintain diverse populations of strategies that capture non-transitive interactions. Policy Space Response Oracles (PSRO) address these issues by iteratively expanding a restricted game with approximate best responses (BRs), yet per-agent BR training makes it prohibitively expensive in many-agent or simulator-expensive settings. We introduce *Joint Experience Best Response (JBR)*, a drop-in modification to PSRO that collects trajectories once under the current meta-strategy profile and reuses this joint dataset to compute BRs for all agents simultaneously. This amortizes environment interaction and improves the *sample efficiency of best-response computation*. Because JBR converts BR computation into an offline RL problem, we propose three remedies for distribution-shift bias: (i) *Conservative JBR* with safe policy improvement, (ii) *Exploration-Augmented JBR* that perturbs data collection and admits theoretical guarantees, and (iii) *Hybrid BR* that interleaves JBR with periodic independent BR updates. Across benchmark multi-agent environments, Exploration-Augmented JBR achieves the best accuracy–efficiency trade-off, while Hybrid BR attains near-PSRO performance at a fraction of the sample cost. Overall, JBR makes PSRO substantially more practical for large-scale strategic learning while preserving equilibrium robustness.

## KEYWORDS

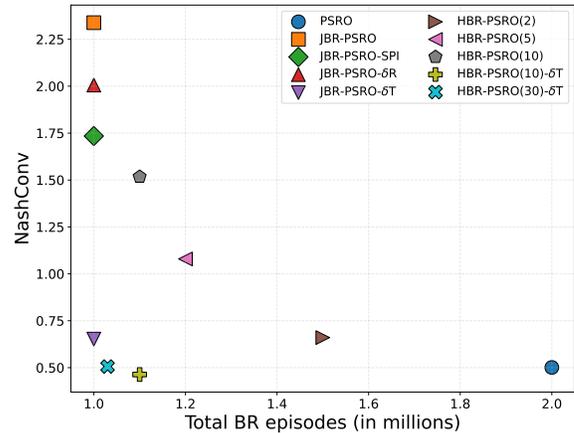
Multi-Agent Reinforcement Learning, Policy Space Response Oracles, Game Theory

### ACM Reference Format:

Ariyan Bighashdel, Thiago D. Simão, and Frans A. Oliehoek. 2026. Sample-Efficient Policy Space Response Oracles with Joint Experience Best Response. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026)*, Paphos, Cyprus, May 25 – 29, 2026, IFAAMAS, 9 pages. <https://doi.org/10.65109/>

## 1 INTRODUCTION

Multi-agent systems (MAS) underlie many real-world domains, from autonomous traffic control to electronic markets, where agents must continually adapt to one another’s strategies. Game theory



**Figure 1: Sample-efficiency – accuracy trade-off in Leduc Poker. Shown are total best-response episodes (in millions,  $x$ -axis) versus minimum NashConv after 100 iterations ( $y$ -axis). Standard PSRO is accurate but requires the highest BR sample cost. Joint Experience Best Response (JBR-PSRO) and its enhanced variants— conservative (JBR-PSRO-SPI), exploration-augmented (JBR-PSRO- $\delta$ R, JBR-PSRO- $\delta$ T), and hybrid (HBR-PSRO(10/30)- $\delta$ T)—drastically reduce the number of BR episodes needed for convergence, with JBR-PSRO- $\delta$ T achieving the best trade-off and hybrid versions approaching PSRO-level accuracy.**

offers strong foundations for analyzing such interactions through equilibrium concepts, but computing equilibria in large or sequential games is infeasible in practice due to the combinatorial growth of policy spaces [23, 30]. This motivates the need for scalable learning-based approaches.

Multi-agent reinforcement learning (MRL) provides one such alternative, enabling agents to improve their policies through repeated interaction [1, 25]. However, MRL faces two key challenges. First, independent learning induces non-stationarity: as each agent updates, the environment faced by others keeps changing [29]. Centralized Training with Decentralized Execution (CTDE) [7, 17] partly mitigates this issue by allowing agents to learn with access to joint observations or gradients during training while acting independently at test time. However, residual instability often persists. Second, many games are non-transitive—no strategy is globally



This work is licensed under a Creative Commons Attribution International 4.0 License.

dominant—and robustness requires maintaining diverse populations of strategies [2, 3, 5, 22].

To address these challenges, the *Policy Space Response Oracles* (PSRO) framework [11] emerged as a synthesis of ideas from several research communities: from *planning*, it builds on the Double Oracle algorithm [20]; from *co-evolution*, it adopts mechanisms such as Nash memory [6] and Parallel Nash Memory [21]; from *empirical game-theoretic analysis* (EGTA), it incorporates simulation-based restricted games [24]; and from *reinforcement learning*, it leverages RL-based oracles to train best responses in high-dimensional sequential environments. At its core, PSRO maintains a restricted game consisting of a subset of strategies, computes a *meta-strategy* (a probability distribution over the restricted strategies) using a chosen solver, and expands the restricted game by adding new (approximate) best responses, i.e., strategies that maximize an agent’s payoff against the current meta-strategy profile (the mixed distribution of the other agents’ strategies). By repeating this process, PSRO incrementally approximates equilibria while preserving diverse strategy populations that can capture non-transitive dynamics. The overall procedure is summarized in Algorithm 1.

Despite its strengths, PSRO suffers from a major limitation: its computational cost. At each iteration, every agent must independently compute a best response, requiring as many reinforcement learning problems as there are agents, and in each case, the remaining agents must be simulated. In domains with many players or costly simulators, this process quickly becomes impractical. Existing works have sought to improve efficiency through parallelization [11, 19], sample-efficient simulation [27, 31], or transfer learning [15, 16, 26]. While such methods reduce wall-clock training time, they do not alter the fundamental structure of PSRO: separate best-response computation for each agent at every iteration.

We propose *Joint Experience Best Response* (JBR), a drop-in modification to PSRO that reuses joint experience across agents. Instead of training each best response independently, JBR collects trajectories once under the current meta-strategy profile and computes best responses for all agents from this shared dataset. This dramatically improves the *sample efficiency of best-response computation*, making PSRO practical in multi-agent or simulator-expensive domains.

Because JBR reuses data generated under the current meta-strategy rather than interacting with the environment online, best-response computation becomes an *offline reinforcement learning* (*offline RL*) problem, where policies must be optimized from a fixed dataset instead of new environment rollouts. This offline setting can introduce bias due to distribution shift. We introduce three complementary variants: (i) *Conservative JBR*, which enforces Safe Policy Improvement [12] to guarantee no degradation below the baseline meta-strategy; (ii) *Exploration-Augmented JBR*, which perturbs data collection to improve coverage while preserving theoretical guarantees; and (iii) *Hybrid BR*, which alternates between JBR and periodic independent BR to combine efficiency with accuracy (Figure 1).

Our findings show that exploration-augmented JBR with targeted perturbation (**JBR-PSRO- $\delta$ T**) achieves near-PSRO equilibrium accuracy at a fraction of the sample cost, while hybrid variants (**HBR-PSRO(k)- $\delta$ T**) reach PSRO-level performance with only minor additional training. Across discrete and continuous environments, PSRO-style methods—including our JBR variants—consistently outperform standard MARL paradigms such as independent learning

and CTDE, confirming that strategically grounded training remains more robust when made sample-efficient.

Our contributions are threefold:

- (1) We introduce *Joint Experience Best Response* (JBR), an efficient modification of PSRO that **improves the sample efficiency of best-response computation** by reusing shared trajectories across agents.
- (2) We identify and address the offline RL limitations of JBR, proposing and evaluating three remedies: Conservative JBR, Exploration-Augmented JBR (with theoretical guarantees), and Hybrid BR.
- (3) We provide a comprehensive empirical study comparing JBR variants not only against Independent BR (standard PSRO), but also against mainstream MARL baselines such as Independent Learning and CTDE, showing that PSRO-based approaches consistently yield superior strategic robustness.

## 2 RELATED WORK

A wide range of extensions to PSRO have been proposed to improve different aspects of the framework, including scalability, exploration, and equilibrium computation. These directions are reviewed in a recent survey [4]. In this paper, we focus specifically on methods aimed at improving the *efficiency* of PSRO.

*Parallelization.* Early work explored reducing wall-clock time through parallel training of best responses. The Deep Cognitive Hierarchy (DCH) [11] introduced a hierarchical schedule in which each player trains against equilibria of lower levels, effectively reusing earlier computations. Pipeline PSRO (P2SRO) [19] refined this idea by coordinating multiple strategies trained in parallel and dynamically optimizing the hierarchy itself. While these methods improve throughput, they require substantial computational resources and do not reduce the overall number of best responses.

*Sample efficiency.* Another line of research targets the reliance on costly simulation. Smith and Wellman [27] proposed to jointly learn a generative model of the game while running PSRO, enabling queries to the learned model instead of the environment. Zhou et al. [31] developed EPSRO, which avoids redundant restricted-game evaluations by exploiting cases where best-response target profiles can be derived without full simulation. Although these approaches cut down on simulator calls, their benefits depend heavily on the quality of the learned model, and the cost of retraining best responses remains unchanged.

*Transfer learning.* A third approach is to accelerate adaptation by reusing knowledge across best responses. NeuPL [15] introduced parameter sharing across strategies in a single network, while Liu et al. [14] trained against sampled mixtures to provide robustness to diverse opponent profiles. Later, Liu et al. [16] combined NeuPL with JPSRO to extend transfer benefits to correlated-equilibrium computation. Smith et al. [26] proposed a Mixed-Oracle method that constructs new best responses by mixing value functions pre-trained against pure strategies. These methods accelerate learning through knowledge reuse, but their effectiveness is tied to how well information transfers across opponents, and they still instantiate new best responses one at a time.

*Summary.* Prior work demonstrates that PSRO can be made more practical through parallelization, efficient simulation, and transfer across best-response computations. While such methods reduce

wall-clock training time, they largely remain implementation-level optimizations: the core PSRO loop continues to generate a separate best response for each agent at every iteration. In contrast, our proposed JBR framework changes this structure directly by reusing shared interaction experience to compute best responses for all agents simultaneously, offering a different route toward scalability.

### 3 BACKGROUND

The core idea of PSRO is to approximate otherwise intractable games by iteratively constructing a smaller *restricted game*, in which each player has access only to a subset of strategies. This restricted game is gradually expanded by adding new best responses, which are computed using RL. In this way, PSRO maintains a tractable approximation of the full game while progressively refining the strategic coverage. Formally, consider an  $n$ -player Markov (stochastic) game defined by:

- a state space  $\mathcal{S}$ ,
- action spaces  $\mathcal{A}_i$  for each player  $i \in N = \{1, \dots, n\}$ , with joint action space  $\mathcal{A} = \times_{i=1}^n \mathcal{A}_i$ ,
- a transition kernel  $P(s' \mid s, a_1, \dots, a_n)$ ,
- player-specific reward functions  $r_i(s, a_1, \dots, a_n)$ ,
- a discount factor  $\gamma \in [0, 1)$ ,
- and an initial state distribution  $\rho_0$ .

A strategy for player  $i$  is a policy  $\pi_i \in \Pi_i$ , i.e., a mapping from states to distributions over  $\mathcal{A}_i$ . Given a policy profile  $\pi = (\pi_1, \dots, \pi_n)$ , the utility of player  $i$  is the expected discounted return of per-step rewards:

$$u_i(\pi) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r_i(s_t, a_{1,t}, \dots, a_{n,t}) \mid \pi \right].$$

PSRO can thus be understood as operating on a *normal-form game induced by the Markov game*,  $\mathcal{G} = (N, (\Pi_i), (u_i))$ , where the pure strategies are full policies and payoffs are their expected returns. In practice, this induced game is intractable because each  $\Pi_i$  is enormous. PSRO therefore maintains restricted policy sets  $X_i \subseteq \Pi_i$ , consisting of the policies generated so far, and defines the restricted empirical game,  $\hat{\mathcal{G}} = (N, (X_i), (\hat{u}_i))$ , where  $\hat{u}_i$  are payoff estimates obtained from rollouts of profiles  $\pi \in X = \times_i X_i$ . The restricted empirical game serves as the basis for strategy updates, representing a smaller, simulation-based approximation of the full game that captures interactions among the currently known policies. The PSRO algorithm alternates between two key components:

- **Meta-Strategy Solver (MSS):** Given  $\hat{\mathcal{G}}$ , the MSS computes a meta-strategy profile (mixed strategy)  $\sigma = (\sigma_i)_{i \in N}$ , where each  $\sigma_i$  is a probability distribution over  $X_i$  that specifies how frequently each policy in the agent’s current set is played. Typical choices include Nash equilibrium solvers or regret-minimization dynamics.
- **Response Oracle (RO):** For each player  $i$ , the RO computes an approximate best-response policy  $\pi_i^{\text{BR}}$  against  $\sigma_{-i}$ . Formally, the best-response operator is

$$br_i(\sigma_{-i}) = \arg \max_{\pi_i \in \Pi_i} u_i(\pi_i, \sigma_{-i}).$$

In practice, the RO is *implemented by deep RL*: the agent is trained in the induced MDP where opponents follow  $\sigma_{-i}$ ,

---

#### Algorithm 1 Policy Space Response Oracles (PSRO)

---

**Require:** Initial restricted policy sets  $(X_i)$

- 1: Estimate  $\hat{\mathcal{G}} = (N, (X_i), (\hat{u}_i))$
  - 2: Initialize  $\sigma \leftarrow \text{MSS}(\hat{\mathcal{G}})$
  - 3: **while** not terminated **do**
  - 4:   **for** player  $i \in N$  **do**
  - 5:     Compute  $\pi_i^{\text{BR}} \leftarrow \text{RO}(\sigma_{-i})$
  - 6:     Update  $X_i \leftarrow X_i \cup \{\pi_i^{\text{BR}}\}$
  - 7:   **end for**
  - 8:   Re-estimate  $\hat{\mathcal{G}}$
  - 9:   Update  $\sigma \leftarrow \text{MSS}(\hat{\mathcal{G}})$
  - 10: **end while**
  - 11: **return**  $\sigma$
- 

yielding a policy  $\pi_i^{\text{BR}}$  that approximates  $br_i(\sigma_{-i})$ . The new policy is then added to  $X_i$ , expanding the restricted game.

The complete procedure is shown in Algorithm 1. The process continues until a stopping criterion is met, often based on *regret*. The regret of player  $i$  under a meta-strategy profile  $\sigma$  is

$$\rho_i^{\mathcal{G}}(\sigma) = \max_{\pi_i \in \Pi_i} u_i(\pi_i, \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}), \quad (1)$$

which measures the potential gain from deviating unilaterally. At a Nash equilibrium, all regrets vanish. The aggregate regret,

$$\rho^{\mathcal{G}}(\sigma) = \sum_{i \in N} \rho_i^{\mathcal{G}}(\sigma), \quad (2)$$

is known as *NashConv* [11] and, in two-player games, corresponds to exploitability. This quantity is widely used to track convergence in PSRO experiments. In practice, it can only be computed exactly in small discrete games where exhaustive best-response search is feasible; in larger or continuous environments, it must be approximated using learned best responses (see Section 5).

## 4 METHOD

Throughout this section we adopt a unified perspective: best responses are derived via *model-based reinforcement learning*, specifically value iteration. This choice is made purely for clarity in explaining the different approaches. In practice, the RO is typically realized by deep RL methods, which approximate the same objective when exact model-based computation is infeasible. All formulations are therefore presented in this model-based framework for consistency, while our experiments employ both model-based and deep RL implementations depending on the environment.

### 4.1 Induced MDP

In PSRO, the best response for player  $i$  is obtained by *reducing* the multi-agent stochastic game to a single-agent MDP: we fix the opponents’ meta-strategy profile, absorb their behavior into the environment, and solve the resulting induced MDP.

PSRO assumes that the game has *perfect recall*, meaning that each agent remembers the entire sequence of states and actions it has observed and taken so far. This assumption ensures that no relevant information is forgotten when making decisions over time. Under perfect recall, Kuhn’s theorem [8] states that any mixed strategy (a distribution over full policies) is realization-equivalent

to a behavior strategy that specifies, at each state, a distribution over available actions. In other words, while playing against a mixed strategy formally induces a POMDP—since the opponent’s policy index is unobserved—the assumption of perfect recall implies that the state  $s$  already encapsulates the relevant history  $h$ . Hence, the two representations are equivalent, and without loss of generality the opponents’ meta-strategy  $\sigma_{-i}$  can be represented as a behavior strategy  $\sigma_{-i}(a_{-i} | s)$ .<sup>1</sup>

Formally, fixing  $\sigma_{-i}$  yields effective transition and reward functions for player  $i$  by marginalizing over the opponents’ actions:

$$P^{\sigma_{-i}}(s' | s, a_i) = \sum_{a_{-i} \in \mathcal{A}_{-i}} \sigma_{-i}(a_{-i} | s) P(s' | s, a_i, a_{-i}), \quad (3)$$

$$r_i^{\sigma_{-i}}(s, a_i) = \sum_{a_{-i} \in \mathcal{A}_{-i}} \sigma_{-i}(a_{-i} | s) r_i(s, a_i, a_{-i}). \quad (4)$$

(Replace sums by integrals for continuous  $\mathcal{A}_{-i}$ .) This defines the induced single-agent MDP for player  $i$ :

$$\mathcal{M}_i^{\sigma_{-i}} = (\mathcal{S}, \mathcal{A}_i, P^{\sigma_{-i}}, r_i^{\sigma_{-i}}, \gamma, \rho_0).$$

## 4.2 Independent Best Response

Let  $Q_i(s, a_i)$  denote the action-value function of player  $i$  in the induced MDP  $\mathcal{M}_i^{\sigma_{-i}}$ . Value iteration computes  $Q_i$  by repeatedly applying the Bellman optimality update:

$$Q_i^{k+1}(s, a_i) = r_i^{\sigma_{-i}}(s, a_i) + \gamma \sum_{s' \in \mathcal{S}} P^{\sigma_{-i}}(s' | s, a_i) \max_{a'_i \in \mathcal{A}_i} Q_i^k(s', a'_i).$$

As  $k \rightarrow \infty$ , the sequence converges to the optimal action-value function  $Q_i^*$ . A greedy policy with respect to  $Q_i^*$ ,

$$\pi_i^{\text{BR}}(s) \in \arg \max_{a_i \in \mathcal{A}_i} Q_i^*(s, a_i),$$

is then a best response to  $\sigma_{-i}$ .

## 4.3 Joint Experience Best Response

While Independent Best Response (§4.2) is the standard approach in PSRO, it has a key drawback: each agent’s best response is trained separately. This creates substantial redundancy: when training one agent, the interaction experience generated cannot be reused by others, leading to repeated and costly environment simulations. In domains with many agents or expensive dynamics, this per-agent training quickly becomes impractical.

To address this inefficiency, we propose *Joint Experience Best Response (JBR)*. The core idea is to collect trajectories once, under the current meta-strategy profile  $\sigma = (\sigma_j)_{j \in N}$ , and reuse this joint dataset to compute best responses for all agents simultaneously. In this setting, all players act according to their meta-strategies during data collection, so the dataset  $\mathcal{D}^\sigma$  consists of trajectories

$$\mathcal{D}^\sigma = \{(s_t, a_{1,t}, \dots, a_{n,t}, r_{1,t}, \dots, r_{n,t}, s_{t+1})\}_{t=0}^T$$

generated by the Markov game when the joint policy profile  $\sigma$  is executed. The overall procedure is summarized in Algorithm 2.

<sup>1</sup>*Remark:* If opponents’ meta-strategies are interpreted as distributions over full policies and one such policy is sampled at the start of an episode, then from the viewpoint of player  $i$  the opponent policy index acts as a hidden variable, inducing a POMDP. Two standard resolutions exist: (i) represent the state space  $\mathcal{S}$  as the set of complete histories, making the process Markov, or (ii) assume *perfect recall*, which by Kuhn’s theorem guarantees that every mixed strategy is realization-equivalent to a behavior strategy. In both cases, we may treat the meta-strategy  $\sigma_{-i}$  as a behavior strategy  $\sigma_{-i}(a_{-i} | s)$ , yielding the induced MDP.

---

## Algorithm 2 PSRO with Joint Experience Best Response

---

**Require:** Initial restricted policy sets  $(X_i)$

- 1: Estimate  $\hat{\mathcal{G}} = (N, (X_i), (\hat{u}_i))$
  - 2: Initialize  $\sigma \leftarrow \text{MSS}(\hat{\mathcal{G}})$
  - 3: **while** not terminated **do**
  - 4:   Collect joint dataset  $\mathcal{D}^\sigma$  under meta-strategy  $\sigma$
  - 5:   **for** player  $i \in N$  **do**
  - 6:     Compute  $\pi_i^{\text{BR}} \leftarrow \text{RO}(\mathcal{D}_{\sigma_{-i}})$
  - 7:     Update  $X_i \leftarrow X_i \cup \{\pi_i^{\text{BR}}\}$
  - 8:   **end for**
  - 9:   Re-estimate  $\hat{\mathcal{G}}$
  - 10:   Update  $\sigma \leftarrow \text{MSS}(\hat{\mathcal{G}})$
  - 11: **end while**
  - 12: **return**  $\sigma$
- 

## 4.4 Naïve JBR

The most direct approach is to treat  $\mathcal{D}^\sigma$  as a fixed offline dataset and then, for each agent  $i$ , construct its induced MDP  $\mathcal{M}_i^{\sigma_{-i}}$  as in §4.1. Best responses are then obtained by applying model-based value iteration on  $\mathcal{M}_i^{\sigma_{-i}}$ , using  $\mathcal{D}^\sigma$  to estimate the transition dynamics  $P^{\sigma_{-i}}$  and rewards  $r_i^{\sigma_{-i}}$ . Formally, the  $Q$ -iteration update is

$$Q_i^{k+1}(s, a_i) = \hat{r}_i^{\sigma_{-i}}(s, a_i) + \gamma \sum_{s' \in \mathcal{S}} \hat{P}^{\sigma_{-i}}(s' | s, a_i) \max_{a'_i} Q_i^k(s', a'_i),$$

where  $\hat{r}_i^{\sigma_{-i}}$  and  $\hat{P}^{\sigma_{-i}}$  are the reward and transition models estimated from  $\mathcal{D}^\sigma$ , respectively. The resulting greedy policy  $\pi_i^{\text{BR}}$  is then added to the restricted game, exactly as in standard PSRO.

Although conceptually simple, this *Naïve JBR* suffers from a critical limitation: since the dataset  $\mathcal{D}^\sigma$  is generated under the current meta-strategy, it may not cover the state–action regions relevant for computing the true best response. This distribution shift makes the problem an instance of offline RL, potentially leading to suboptimal approximations of best responses. To address these limitations, we introduce three complementary remedies. The first is a conservative safeguard based on Safe Policy Improvement (§4.5), which ensures that policies do not underperform the current meta-strategy. The second is an exploration-based approach (§4.6), which perturbs the meta-strategy during data collection to improve coverage and admits theoretical guarantees. Finally, a hybrid variant (§4.7) alternates between joint and independent best responses to balance efficiency and accuracy. Together, these three approaches mitigate the shortcomings of Naïve JBR.

## 4.5 Conservative JBR

To address the offline RL bias of Naïve JBR, we adopt the principle of *Safe Policy Improvement (SPI)* [12]. The idea is to restrict the learned policy to only deviate from its baseline  $\sigma_i$  in regions of the state–action space that are sufficiently well supported by  $\mathcal{D}^\sigma$ . Wherever the dataset lacks coverage, the policy is forced to fall back to  $\sigma_i$ , guaranteeing that performance cannot degrade below the baseline.

Formally, let  $N_{\mathcal{D}}(s, a_i)$  denote the number of times  $(s, a_i)$  appears in  $\mathcal{D}^\sigma$ , and fix a threshold  $N_\wedge$ . We define the set of *uncertain* state–action pairs as

$$B_i = \{(s, a_i) \mid N_{\mathcal{D}}(s, a_i) < N_\wedge\}.$$

The SPI constraint then requires that for all  $(s, a_i) \in B_i$ ,

$$\pi_i(a_i | s) = \sigma_i(a_i | s).$$

That is, in poorly covered regions, the new policy exactly reproduces the baseline meta-strategy  $\sigma_i$ . Outside  $B_i$ , the policy is optimized via offline value iteration on  $\mathcal{M}_i^{\sigma_{-i}}$ , using  $\mathcal{D}^\sigma$  to estimate transitions and rewards.

The resulting policy  $\pi_i^{\text{SPI}}$  is not necessarily a best response to  $\sigma_{-i}$ , but it is guaranteed to be no worse than  $\sigma_i$  (up to estimation error), while strictly improving wherever  $\mathcal{D}^\sigma$  provides sufficient coverage. Thus,  $\pi_i^{\text{SPI}}$  can safely be added to the restricted policy set in PSRO. However, as we show in our experiments, this conservative approach, while outperforming Naïve JBR, still performs poorly compared to the original PSRO with independent best responses. This motivates the need for a more effective remedy.

## 4.6 Exploration-Augmented JBR

While Conservative JBR prevents performance degradation, it remains overly cautious: improvements are confined to regions already well represented in the dataset, limiting its ability to match the performance of independent best responses. To overcome this, we propose to deliberately broaden coverage during data collection by perturbing the meta-strategy with exploration.

Let  $\sigma = (\sigma_j)_{j \in N}$  be the current meta-strategy profile. During data collection we use an  $\delta$ -perturbed meta-strategy that keeps agents mostly on-policy w.r.t.  $\sigma$  but injects exploration with probability  $\delta$ . For each agent  $j$ , define a per-state mixed behavior

$$\tilde{\sigma}_j^{(\delta, v_j)}(a_j | s) = (1 - \delta) \sigma_j(a_j | s) + \delta v_j(a_j | s), \quad (5)$$

where  $v_j$  is an exploration policy. Agents act independently according to  $\tilde{\sigma}^{(\delta, v)} = (\tilde{\sigma}_j^{(\delta, v_j)})_{j \in N}$ , generating a joint dataset  $\mathcal{D}^{\tilde{\sigma}}$ .

We consider two concrete choices, yielding two exploration strategies:

**4.6.1  $\delta$ -random exploration.** Set  $v_j^{\text{rand}}(a_j | s) = \text{Unif}(\mathcal{A}_j)$ , i.e.,

$$\tilde{\sigma}_j^{(\delta, \text{rand})}(a_j | s) = (1 - \delta) \sigma_j(a_j | s) + \delta \frac{1}{|\mathcal{A}_j|}.$$

This injects state-agnostic coverage while keeping the process mostly on-policy w.r.t.  $\sigma$ .

**4.6.2  $\delta$ -targeted exploration.** Let  $\pi_j^{\text{BR,cur}}$  denote the current BR candidate for agent  $j$  (the policy being trained in JBR this iteration). Set  $v_j^{\text{tgt}}(a_j | s) = \pi_j^{\text{BR,cur}}(a_j | s)$ , i.e.,

$$\tilde{\sigma}_j^{(\delta, \text{tgt})}(a_j | s) = (1 - \delta) \sigma_j(a_j | s) + \delta \pi_j^{\text{BR,cur}}(a_j | s).$$

This concentrates additional samples near currently promising BR behaviors, improving value estimation where it matters for BR computation.

**4.6.3 Theoretical Implication.** While exploration broadens coverage and mitigates offline RL bias, it also changes the learning target: agents no longer best respond to the exact meta-strategy  $\sigma$ , but rather to its  $\delta$ -perturbed variant  $\tilde{\sigma}$ . This raises the question of whether PSRO with exploration still preserves its theoretical guarantees.

We now analyze this effect. In particular, we show that even though exploration changes the response oracle’s target distribution, PSRO with exploration-augmented JBR still preserves its convergence guarantees up to a bounded error. We focus on finite two-player zero-sum Markov games with bounded payoffs in  $[\underline{u}, \bar{u}]$  and range  $R = \bar{u} - \underline{u}$ .

**THEOREM 4.1 (EXPLORATION-AUGMENTED JBR).** *Let  $\sigma$  be the current meta-strategy profile and  $\tilde{\sigma}$  its  $\delta$ -perturbed variant used for data collection. If each agent computes an  $\varepsilon$ -best response to  $\tilde{\sigma}$ , then upon termination the resulting meta-strategy is an  $(\varepsilon + 2R\delta)$ -Nash equilibrium of the original game.*

*Proof sketch.* Payoffs are linear in mixed strategies, so perturbing the opponents’ meta-strategy by  $\delta$  changes the payoff of any policy by at most  $2R\delta$ . Hence a best response to  $\tilde{\sigma}$  is also an approximate best response to  $\sigma$ , up to this error. Since PSRO’s termination check is performed under  $\sigma$ , no profitable deviations are missed, giving the stated  $(\varepsilon + 2R\delta)$  bound.

*Implication.* This result shows that exploration-augmented JBR preserves the convergence guarantees of PSRO up to a controlled error: the deviation from equilibrium grows only linearly with  $\delta$ , while the added exploration improves coverage and reduces offline RL bias. It should be noted, however, that the guarantee holds only under the assumption that an  $\varepsilon$ -best response to  $\tilde{\sigma}$  can be found. In practice, computing such a response from a fixed dataset is not guaranteed, as limited coverage may prevent reliable policy improvement. This motivates the use of multiple exploration strategies, which aim to enrich the dataset and increase the likelihood that a sufficiently good response can be learned in practice.

## 4.7 Hybrid Best Response

While JBR substantially reduces simulation cost, its reliance on offline data introduces approximation errors. Variants such as Conservative JBR and Exploration-Augmented JBR mitigate these errors, but they may still deviate from the accuracy of Independent BR (IBR), which computes exact best responses at each iteration. This raises the question of whether we can combine the efficiency of JBR with the accuracy of IBR.

We therefore consider a *Hybrid BR* approach that alternates between the two. Specifically, JBR (possibly in one of its variants) is used in most iterations, while every  $k$ -th iteration is replaced by a standard Independent BR update. Formally, let  $t$  denote the iteration counter. If  $t \bmod k = 0$ , each agent  $i$  computes a best response  $\pi_i^{\text{IBR}}$  via independent training against  $\sigma_{-i}$ ; otherwise, responses are obtained using JBR (e.g., Naïve, Conservative, or Exploration-Augmented).

This hybrid strategy retains the sample efficiency benefits of JBR in the majority of iterations, while periodically correcting approximation errors by injecting accurate best responses from IBR. In our experiments, we evaluate whether such periodic corrections are sufficient to close the performance gap between JBR-based variants and original PSRO.

## 5 EXPERIMENTS

The aim of our experiments is to assess the *feasibility and reliability* of the proposed JBR framework. By design, JBR computes best

responses for all agents simultaneously from a single shared dataset, which trivially improves sample efficiency by reducing environment interactions. The central question, however, is whether JBR can still produce accurate best responses—and thus preserve PSRO’s convergence behavior—when data are reused across agents.

## 5.1 Experimental Setup

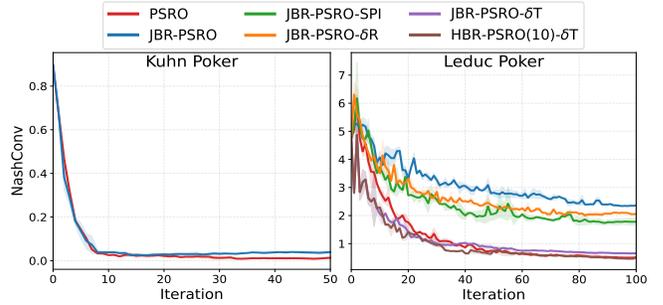
*Games.* We consider two classic two-player zero-sum poker games, *Kuhn Poker* [9] and *Leduc Poker* [28], followed by three continuous multi-agent particle environments [17]: *Simple Tag*, *Simple Adversary*, and *Simple Push*. This sequence increases the difficulty of learning reliable best responses: Kuhn Poker is small and fully covered by the meta-strategy, while Leduc Poker introduces imperfect information and a larger state space that exposes offline-learning bias. The particle environments further stress-test the method under continuous control and partial observability, where joint data reuse may degrade policy quality.

*Methods.* We evaluate the following algorithms:

- **PSRO**: standard PSRO with independent best responses (baseline);
- **JBR-PSRO**: naïve joint-experience variant computing all BRs from shared data;
- **JBR-PSRO-SPI**: conservative JBR using Safe Policy Improvement (SPI). The SPI variant requires a coverage threshold  $N_\lambda$  to determine when to revert to the baseline meta-strategy. To isolate the method’s intrinsic potential rather than its sensitivity to this hyperparameter, we employ an oracle-tuned setting that iterates over  $N_\lambda \in [0, 50]$  at each iteration and selects the best-performing value. While not practical, this allows us to evaluate the idealized upper bound of SPI performance;
- **JBR-PSRO- $\delta R$** : exploration-augmented JBR with  $\delta$ -random exploration;
- **JBR-PSRO- $\delta T$** : exploration-augmented JBR with  $\delta$ -targeted exploration;
- **HBR-PSRO( $k$ )**: hybrid variant alternating between JBR and IBR every  $k$  iterations, which can also be combined with exploration (e.g., **HBR-PSRO( $k$ )- $\delta T$** ).

For comparison with standard MARL paradigms, we also include **IL** (independent learners using DDPG [13]) and **CTDE** (centralized training with decentralized execution using MADDPG [17]).

*Implementation details.* All PSRO variants use the same meta-strategy solver, namely *projected replicator dynamics* [11], to compute the Nash equilibrium of the empirical payoff matrix at each iteration. In the poker domains, best responses (BRs) are computed via model-based value iteration, while in the particle environments we employ deep RL oracles implemented with DDPG or MADDPG, depending on whether training is independent or centralized. Each iteration adds one new policy per agent and updates the meta-strategy accordingly. Although some particle environments involve more than two individual agents—such as *Simple Tag* (one runner vs. two adversaries) and *Simple Adversary* (two cooperating agents vs. one adversary)—we treat them as two-player games by grouping agents into two opposing teams: *agents* and *adversaries*. This team-based formulation follows the *Team PSRO* framework [18], ensuring



**Figure 2: Convergence of PSRO and JBR in two poker games of increasing complexity.** Left: In *Kuhn Poker*, JBR remains close to PSRO, indicating that it is feasible when the state space is small and well covered. Right: In *Leduc Poker*, naive JBR diverges from PSRO as complexity grows, revealing the offline-learning bias that motivates the JBR variants.

that equilibrium analysis remains well-defined at the team level. For best-response computation, we employ DDPG when the team consists of a single agent and MADDPG when the team includes multiple cooperating agents.

Following prior work [11, 31], we run PSRO for 100 iterations in the poker games, allocating a budget of 10,000 episodes for computing best responses. In the particle environments, we run 50 iterations with a BR budget of 50,000 episodes per iteration. Importantly, in standard PSRO each agent receives this full budget independently, whereas in JBR all agents *share* the same budget to compute their BRs jointly, reflecting the gain in training efficiency. Unless otherwise stated, the exploration rate  $\delta$  is set to 0.1 for **JBR-PSRO- $\delta R$**  and 0.5 for **JBR-PSRO- $\delta T$** ; sensitivity analyses for these values are reported in Section 5.6. All reported results are averaged over three random seeds.

To evaluate convergence, we track two metrics: (i) *NashConv*, measuring equilibrium accuracy, and (ii) *total BR episodes*, quantifying overall sample efficiency. In small discrete games such as Kuhn and Leduc Poker, NashConv is computed exactly using OpenSpiel’s implementation [10], where a full tree search is performed to obtain the true best response. In continuous particle environments, where exact BRs are intractable, we instead estimate an *approximate NashConv*: for each agent we train three BR candidates with different random initializations and an extended BR budget of 100,000 episodes, and select the best-performing one to approximate the true best response.

## 5.2 Is Joint Experience Best Response Feasible?

We first investigate if JBR can serve as a reliable alternative to independent best responses. We evaluate **PSRO** and **JBR-PSRO** in *Kuhn Poker*, a minimal imperfect-information game with a small and fully covered state space.

As shown in the left panel of Figure 2, the performance gap between JBR-PSRO and PSRO remains small across iterations, indicating that JBR closely follows the convergence behavior of standard PSRO. Because Kuhn Poker’s state space is compact and every state-action pair is visited under the joint meta-strategy, there is no distribution-shift bias when computing best responses from shared

data. This confirms that JBR is *feasible* in low-complexity domains where the induced MDPs are well covered by on-policy trajectories.

### 5.3 When Does Naïve JBR Fail?

We next examine how JBR behaves as best-response computation becomes more complex. To this end, we compare **PSRO** and **JBR-PSRO** in *Leduc Poker*, which features a significantly larger state space and deeper decision structure. Unlike Kuhn Poker, the joint dataset collected under the meta-strategy covers only a subset of reachable trajectories, leading to an offline-learning bias when best responses are computed from shared experience.

As shown in the right panel of Figure 2, the performance gap between JBR-PSRO and PSRO increases considerably: while PSRO continues to reduce NashConv, JBR-PSRO stagnates at higher exploitability. This demonstrates that naïve JBR, although efficient, fails to maintain equilibrium accuracy once the induced MDPs become partially observable or high-dimensional. These results motivate the development of the *JBR variants* discussed next.

### 5.4 How Effective Are the JBR Variants?

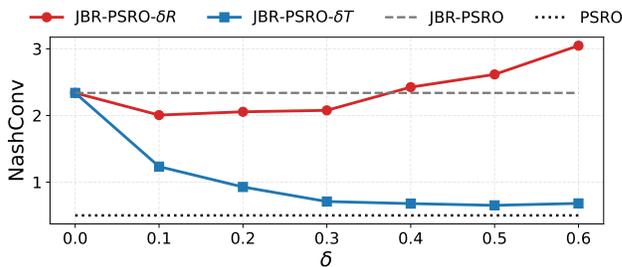
We now evaluate whether the enhanced *JBR variants* can mitigate the offline-learning bias observed in naïve JBR. Our focus is on the performance of **JBR-PSRO-SPI**, **JBR-PSRO- $\delta$ R**, and **JBR-PSRO- $\delta$ T** compared to the baseline **JBR-PSRO**.

As shown in the right panel of Figure 2, all enhanced JBR variants outperform the naïve baseline, confirming that explicit mechanisms for coverage and stability substantially improve best-response quality. However, their relative effectiveness differs: **JBR-PSRO-SPI** is conservative, as its Safe Policy Improvement constraint forces the policy to revert to the baseline meta-strategy in under-sampled regions, limiting progress. **JBR-PSRO- $\delta$ R**, which employs random exploration, increases data diversity but lacks direction, leading to only moderate gains. In contrast, **JBR-PSRO- $\delta$ T** achieves near-PSRO performance by guiding exploration toward promising trajectories. Its  $\delta$ -targeted exploration explicitly conditions on the current best response during data collection—anticipating how opponents’ strategies will change after the next update. This forward-looking behavior resembles *opponent modeling*, enabling more accurate and stable best responses. Interestingly, this forecasting nature also explains the faster convergence observed in early iterations of **JBR-PSRO- $\delta$ T** compared to standard PSRO, as agents proactively adapt to the direction of the upcoming meta-strategy updates.

Based on these observations, we identify **JBR-PSRO- $\delta$ T** as the most effective variant. To further close the small remaining gap to PSRO, we test the hybrid approach that periodically replaces JBR updates with independent best responses. As shown in Figure 2, this configuration, **HBR-PSRO(10)- $\delta$ T**, achieves PSRO-level convergence while preserving the substantial sample-efficiency gains of JBR.

### 5.5 How Sample-Efficient are the JBR variants?

We now analyze the *sample efficiency of best-response computation* across different PSRO methods. Figure 1 reports results in *Leduc Poker*, showing the total number of environment episodes used for best-response training (in millions,  $x$ -axis) versus the minimum



**Figure 3: Effect of the exploration rate  $\delta$  in Leduc Poker. Minimum NashConv after 100 iterations for random (JBR-PSRO- $\delta$ R) and targeted (JBR-PSRO- $\delta$ T) exploration. Random exploration peaks at  $\delta=0.1$  then degrades beyond 0.4, while targeted exploration improves up to  $\delta=0.5$  and remains consistently better than naïve JBR.**

NashConv achieved after 100 iterations ( $y$ -axis). This directly quantifies how efficiently each method computes best responses relative to its equilibrium quality.

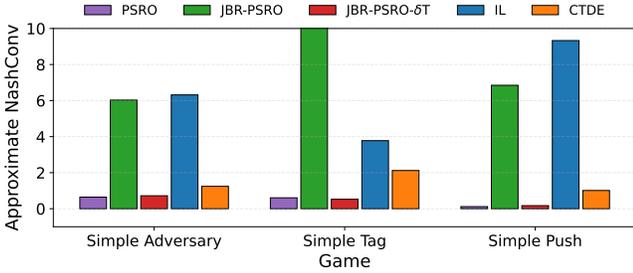
Standard **PSRO** achieves the lowest NashConv but is highly sample-inefficient, as each agent independently trains a new best response at every iteration. In contrast, **JBR-PSRO** and its enhanced variants reuse a single shared dataset to compute best responses for all agents simultaneously, effectively amortizing the simulation cost across agents. This substantially reduces the number of environment interactions required for best-response computation while maintaining competitive equilibrium accuracy.

Among the enhanced variants, **JBR-PSRO- $\delta$ T** achieves the best trade-off between equilibrium accuracy and BR sample efficiency, reaching near-PSRO NashConv with half of PSRO’s total BR episodes. Its hybrid counterparts, **HBR-PSRO(10)- $\delta$ T** and **HBR-PSRO(30)- $\delta$ T**, further close the remaining gap to PSRO by occasionally performing independent BR updates, requiring only a small increase in BR episodes. **JBR-PSRO-SPI** remains more stable but conservative, while **JBR-PSRO- $\delta$ R** provides moderate improvements due to uninformed exploration. Overall, these results confirm that JBR-based training substantially improves the *sample efficiency of best-response computation*, enabling practical deployment of PSRO in domains where environment interaction cost dominates runtime.

### 5.6 Sensitivity to the Exploration Rate $\delta$

We next study how  $\delta$  influences performance in *Leduc Poker*. Figure 3 reports the final NashConv after 100 iterations for both **JBR-PSRO- $\delta$ R** and **JBR-PSRO- $\delta$ T** across different  $\delta$  values.

For **JBR-PSRO- $\delta$ R**, performance initially improves as  $\delta$  increases, reaching its best value at  $\delta = 0.1$ . Beyond this point, higher levels of random mixing gradually degrade performance, and after  $\delta = 0.4$ , results become worse than the naïve **JBR-PSRO** baseline. This behavior reflects the instability introduced by excessive randomization, which distorts value estimates and weakens the best-response signal. In contrast, **JBR-PSRO- $\delta$ T** shows a steady improvement as  $\delta$  increases, plateauing around  $\delta = 0.4$  and peaking near  $\delta = 0.5$ . A slight drop occurs at  $\delta = 0.6$ , but performance remains well above that of **JBR-PSRO** for all tested values. This



**Figure 4: Approximate NashConv in continuous multi-agent environments. Comparison of PSRO, JBR-PSRO, JBR-PSRO- $\delta T$ , and two MARL baselines (IL/DDPG, CTDE/MADDPG) on *Simple Tag*, *Simple Adversary*, and *Simple Push*. PSRO and JBR-PSRO- $\delta T$  achieve comparable approximate NashConv across games; both PSRO methods outperform IL and CTDE.**

robustness indicates that targeted exploration continues to enhance coverage and opponent forecasting over a broad range of  $\delta$ .

Overall, these results confirm that random and targeted exploration behave differently: random exploration requires a small, cautious perturbation to avoid destabilizing learning, whereas targeted exploration benefits from stronger mixing that accentuates the predictive structure of the meta-strategy.

### 5.7 Beyond Poker: Continuous Multi-Agent Environments

We now evaluate whether the conclusions from poker generalize to continuous multi-agent control. In the *Simple Tag*, *Simple Adversary*, and *Simple Push* particle environments, exact best responses are intractable; accordingly, we report an *approximate NashConv*. For each agent, we train three BR candidates with different random initializations under an extended BR budget of 100,000 episodes and select the best-performing candidate to approximate the true BR (the same protocol is used to score IL and CTDE).

Figure 4 compares PSRO, JBR-PSRO, and JBR-PSRO- $\delta T$  (the best-performing enhanced variant), alongside two standard MARL frameworks: IL (DDPG) and CTDE (MADDPG). Across all three environments, PSRO and JBR-PSRO- $\delta T$  attain comparable approximate NashConv, while JBR-PSRO lags behind—mirroring the Leduc findings and underscoring the need for targeted exploration when data are reused. Moreover, both PSRO and JBR-PSRO- $\delta T$  consistently outperform IL and CTDE, indicating greater robustness to strategic non-stationarity. These results reinforce the usefulness of PSRO-style game-theoretic training in continuous domains and highlight why improving the sample efficiency of best-response computation is practically important.

### 5.8 Summary of Empirical Insights

Across all experiments, our results collectively demonstrate that JBR substantially improves the *sample efficiency of best-response computation* in PSRO while preserving strong equilibrium performance when paired with targeted exploration. In small, fully covered games (*Kuhn Poker*), JBR matches PSRO’s convergence behavior, establishing feasibility under well-sampled conditions. In larger

games (*Leduc Poker*), naïve JBR underperforms due to offline bias, but enhanced variants—especially JBR-PSRO- $\delta T$ —restore accuracy by anticipating opponent updates; HBR-PSRO( $k$ )- $\delta T$  closes the remaining gap to PSRO with only a modest increase in BR episodes. Ablations show that random exploration requires small mixing, whereas targeted exploration benefits from stronger perturbation (optimal around  $\delta=0.5$ ). In continuous multi-agent control, JBR-PSRO- $\delta T$  again achieves performance comparable to PSRO while being far more sample-efficient. Moreover, both PSRO and JBR-PSRO- $\delta T$  demonstrate greater robustness than standard MARL frameworks such as IL and CTDE, highlighting the importance of PSRO-style game-theoretic training for stable coordination—and showing that improving the efficiency of best-response computation makes such training practically viable in larger-scale domains.

## 6 DISCUSSION

*Practical implications.* PSRO remains one of the most reliable approaches for learning strategically robust policies, yet its use is often constrained by the high cost of best-response computation. The proposed JBR framework mitigates this limitation by reusing experience across agents, with JBR-PSRO- $\delta T$  providing the best trade-off between equilibrium accuracy and BR sample efficiency. When near-exact PSRO accuracy is required, hybrid updates (HBR-PSRO( $k$ )- $\delta T$ ) offer a simple way to recover full performance with minimal additional cost.

*Interpretation.* Targeted exploration acts as an implicit form of opponent modeling: by conditioning data collection on the current best response, agents forecast likely opponent adaptations, explaining the faster early convergence of JBR-PSRO- $\delta T$ . This connection highlights how predictive exploration can improve both efficiency and stability in meta-strategy learning.

*Limitations.* Despite the gains, PSRO and JBR variants still require many iterations and substantially more samples than typical MARL methods. Scalability to larger policy populations and high-dimensional continuous domains remains challenging, as meta-strategy updates and payoff matrix growth introduce additional computational cost. Developing adaptive schedules for  $\delta$  and  $k$ , or integrating learned priors to guide exploration, are promising directions for reducing this overhead.

## 7 CONCLUSION

We introduced the Joint Experience Best Response (JBR) framework, which reduces the cost of best-response computation in PSRO by reusing shared interaction data across agents. While naïve JBR can suffer from offline-learning bias in complex games, the enhanced variant JBR-PSRO- $\delta T$  and its hybrid extension HBR-PSRO( $k$ )- $\delta T$  recover PSRO-level equilibrium accuracy with far greater sample efficiency. Experiments across both discrete and continuous multi-agent domains show that JBR-based training maintains the robustness of PSRO while making it more practical for large-scale applications. Future work will explore adaptive exploration and hybrid schedules, scalable meta-strategy updates, and extensions of targeted exploration to deep continuous settings.

## REFERENCES

- [1] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. 2024. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press. <https://www.marl-book.com>
- [2] Ariyan Bighashdel, Daan de Geus, Pavol Jancura, and Gijs Dubbelman. 2023. Coordinating Fully-Cooperative Agents Using Hierarchical Learning Anticipation. arXiv preprint arXiv:2303.08307.
- [3] Ariyan Bighashdel, Daan De Geus, Pavol Jancura, and Gijs Dubbelman. 2024. Off-policy action anticipation in multi-agent reinforcement learning. *Journal of Machine Learning Research* 25, 67 (2024), 1–31.
- [4] Ariyan Bighashdel, Yongzhao Wang, Stephen McAleer, Rahul Savani, and Frans A. Oliehoek. 2024. Policy Space Response Oracles: A Survey. In *IJCAI*. ijcai.org, 7951–7961.
- [5] Wojciech M. Czarnecki, Gauthier Gidel, Brendan D. Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. 2020. Real world games look like spinning tops. In *NeurIPS*. 17443–17454.
- [6] Sevan G. Ficici and Jordan B. Pollack. 2003. A Game-Theoretic Memory Mechanism for Coevolution. In *GECCO*. 286–297.
- [7] Landon Kraemer and Bikramjit Banerjee. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing* 190 (2016), 82–94.
- [8] Harold W Kuhn. 1953. Extensive games and the problem of information. *Contributions to the Theory of Games* 2, 28 (1953), 193–216.
- [9] Harold W Kuhn. 2016. A simplified two-person poker. *Contributions to the Theory of Games* 1 (2016), 97–103.
- [10] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, et al. 2019. OpenSpiel: A framework for reinforcement learning in games. arXiv preprint arXiv:1908.09453.
- [11] Marc Lanctot, Vinicius Flores 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 *NIPS*. 4190–4203.
- [12] Romain Laroche, Paul Trichelair, and Remi Tachet Des Combes. 2019. Safe policy improvement with baseline bootstrapping. In *ICML*. 3652–3661.
- [13] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *ICLR (Poster)*.
- [14] Siqi Liu, Marc Lanctot, Luke Marris, and Nicolas Heess. 2022. Simplex neural population learning: Any-mixture bayes-optimality in symmetric zero-sum games. In *ICML*. 13793–13806.
- [15] Siqi Liu, Luke Marris, Daniel Hennes, Josh Merel, Nicolas Heess, and Thore Graepel. 2022. NeuPL: Neural population learning. In *ICLR*.
- [16] Siqi Liu, Luke Marris, Marc Lanctot, Georgios Piliouras, Joel Z Leibo, and Nicolas Heess. 2024. Neural Population Learning beyond Symmetric Zero-Sum Games. In *AAMAS*. 1247–1255.
- [17] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *NIPS*. 6379–6390.
- [18] Stephen McAleer, Gabriele Farina, Gaoyue Zhou, Mingzhi Wang, Yaodong Yang, and Tuomas Sandholm. 2023. Team-PSRO for Learning Approximate TMECOR in Large Team Games via Cooperative Reinforcement Learning. (2023), 45402–45418.
- [19] Stephen McAleer, John B. Lanier, Roy Fox, and Pierre Baldi. 2020. Pipeline PSRO: A Scalable Approach for Finding Approximate Nash Equilibria in Large Games. (2020), 20238–20248.
- [20] H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. 2003. Planning in the Presence of Cost Functions Controlled by an Adversary. In *ICML*. 536–543.
- [21] Frans A. Oliehoek, Edwin D. de Jong, and Nikos Vlassis. 2006. The parallel Nash Memory for asymmetric games. In *GECCO*. 337–344.
- [22] Ricky Sanjaya, Jun Wang, and Yaodong Yang. 2022. Measuring the non-transitivity in chess. *Algorithms* 15, 5 (2022), 152.
- [23] Rahul Savani and Theodore L. Turocy. 2023. Gambit: The Package for doing Computation in Game Theory, Version 16.1.0. <http://www.gambit-project.org>
- [24] L. Julian Schwartzman and Michael P. Wellman. 2009. Exploring large strategy spaces in empirical game modeling. In *AMEC*. 139–152.
- [25] Yoav Shoham, Rob Powers, and Trond Grenager. 2007. If multi-agent learning is the answer, what is the question? *Artificial intelligence* 171, 7 (2007), 365–377.
- [26] Max Olan Smith, Thomas Anthony, and Michael P Wellman. 2023. Strategic knowledge transfer. *Journal of Machine Learning Research* 24, 233 (2023), 1–96.
- [27] Max Olan Smith and Michael P Wellman. 2023. Co-learning empirical games and world models. arXiv preprint arXiv:2305.14223.
- [28] Finnegan Southey, Michael H. Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and D. Chris Rayner. 2005. Bayes? Bluff: Opponent Modelling in Poker. In *UAI*. 550–558.
- [29] Karl Tuyls and Gerhard Weiss. 2012. Multiagent learning: Basics, challenges, and prospects. *Ai Magazine* 33, 3 (2012), 41–41.
- [30] Bernhard von Stengel. 2002. Computing equilibria for two-person games. In *Handbook of game theory with economic applications*. Vol. 3. Elsevier, 1723–1759.
- [31] Ming Zhou, Jingxiao Chen, Ying Wen, Weinan Zhang, Yaodong Yang, Yong Yu, and Jun Wang. 2022. Efficient policy space response oracles. arXiv preprint arXiv:2202.00633.