Conservative Equilibrium Discovery in Offline Game-Theoretic Multiagent Reinforcement Learning

AAAI Track

Austin A. Nguyen
University of Michigan
Ann Arbor, United States
ngaustin@umich.edu

Michael P. Wellman
University of Michigan
Ann Arbor, United States
wellman@umich.edu

ABSTRACT

Offline learning of strategies takes data efficiency to its extreme by restricting algorithms to a fixed dataset of state-action trajectories. We consider the problem in a mixed-motive multiagent setting, where the goal is to solve a game under the offline learning constraint. We first frame this problem in terms of selecting among candidate equilibria. Since datasets may inform only a small fraction of game dynamics, it is generally infeasible in offline game-solving to even verify a proposed solution is a true equilibrium. Therefore, we consider the relative probability of low regret (i.e., closeness to equilibrium) across candidates based on the information available. Specifically, we extend Policy Space Response Oracles (PSRO), an online game-solving approach, by quantifying game dynamics uncertainty and modifying the RL objective to skew towards solutions more likely to have low regret in the true game. We further propose a novel meta-strategy solver, tailored for the offline setting, to guide strategy exploration in PSRO. Our incorporation of **C**onservatism principles from **Off**lin**e** reinforcement learning approaches for strategy **E**xploration gives our approach its name: COffeE-PSRO. Experiments demonstrate COffeE-PSRO's ability to extract lower-regret solutions than state-of-the-art offline approaches and reveal relationships between algorithmic components, empirical game fidelity, and overall performance.

KEYWORDS

Empirical game-theoretic analysis; game-solving; multiagent reinforcement learning; offline reinforcement learning

ACM Reference Format:
Austin A. Nguyen and Michael P. Wellman. 2026. Conservative Equilibrium Discovery in Offline Game-Theoretic Multiagent Reinforcement Learning: AAAI Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 12 pages. https://doi.org/10.65109/XEYY6214

1 INTRODUCTION

Game-theoretic analysis is a powerful tool when evaluating multiagent systems because it focuses attention on configurations where agents behave rationally relative to each other. In complex games with large strategy spaces, identifying such equilibrium profiles through ***multiagent reinforcement learning*** (MARL) requires substantial computation and *data*. Gathering ample data for game-theoretic analysis can thus be difficult and costly in real-world applications. A capability to perform such analysis *offline* from pre-generated, fixed-size datasets addresses the extreme case of limits on real-time data collection.

Strategy generation using offline reinforcement learning is well-studied in single-agent and cooperative multiagent domains. Many offline learning algorithms adopt the *conservatism principle*: strategies should stay in dataset-informed regions and assume unknown regions yield low returns. This can be interpreted loosely as *policy selection*: among policies that seem approximately optimal, favor ones for which the dataset provides strong evidence and thus better relative confidence of high actual (true) return. Analogously, we frame offline game-solving as *equilibrium selection*, where we choose among profiles apparently in approximate equilibrium, aiming to minimize actual (true-game) regret.

Our contribution is in the application and extension of conservatism to offline *game-solving*. This extension poses a special challenge: consideration of a candidate strategy profile's potential strategic deviations. Our approach achieves this by combining offline RL algorithmic concepts and Policy Space Response Oracles (PSRO) [11], an online game-solving framework that iteratively extends a strategy set using deep RL best-responses to mixed strategies in an empirically estimated game model. Given a dataset of trajectories (sequences of true-game multiagent interactions), we first train an ensemble dynamics model to replace an online simulator and use prediction differences within the ensemble to quantify uncertainty under the dataset. We apply conservatism by introducing a response objective that balances reward, uncertainty given the current response target, and uncertainty of potential strategic deviations. Further, we propose a novel meta-strategy solver that minimizes a heuristic-based, pessimistic estimate of regret. This extension of **C**onservatism from **Off**lin**e** RL approaches to aid strategy **E**xploration in PSRO gives our approach its name: COffeE-PSRO. In experiments on a sequential bargaining game, we find that COffeE-PSRO tends to output lower-regret solutions than tested baselines. Our results highlight the role of conservatism in improving model fidelity, robustness to dataset changes, and solution regret.

2 RELATED WORK

Conservatism is a common theme in offline RL approaches [12]. Model-free methods have incorporated conservatism through policy regularization [8, 30], value penalties [9], and implicit behavioral constraints [17, 19], while model-based methods have used uncertainty quantification and reward penalties [7, 32]. Conservative

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/XEYY6214

methods in offline MARL have addressed centralized coordination [25] and nonstationarity in decentralized optimization [5].

Yang and Wang [31] provide an overview of game-theoretic approaches in MARL. The framework of ***Policy Space Response Oracles*** (PSRO) [2, 11] employs an empirical game modeling approach [29], iteratively extending the strategy set using deep RL to derive policies maximizing return against an other-player ***response target*** defined by a ***meta-strategy solver*** (MSS). Substantial work has investigated strategy exploration [6] in PSRO, in particular through MSS selection of response targets [1, 15, 16, 22, 27] or through definition of response objectives [18, 20, 28].

In offline game-theoretic MARL, theoretical work has investigated the necessary conditions for equilibrium extraction [4, 34], defining *unilateral concentration* as the property where a dataset covers an equilibrium and all its unilateral strategic deviations. Other approaches focus on offline evaluation of given strategies to determine which are most likely equilibria [33, 34]. Shao et al. [21] employ conservative critics and low-rank matrix completions to select equilibria given a fixed dataset of payoff data for select policy profiles. Most relevant here is work on strategy generation for offline equilibrium extraction [14], which uses a trained dynamics model in place of a simulator and mixes the final solution with a behavior cloning policy. Chen et al. [3] adapts fictitious self-play using single-agent offline RL and importance sampling.

3 NOTATION

We model a multiagent environment as a ***stochastic game*** defined by $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, Z, O, n, \gamma \rangle$. The environment follows trajectories of states $s \in \mathcal{S}$, influenced by actions $a \in \mathcal{A}$ taken alternately by each of n agents. Each action prompts a transition according to the state transition function $\mathcal{T}(s' \mid s, a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$. Agent rewards are given by $\mathcal{R}(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^n$, and $\gamma \in [0, 1)$ is a discount factor. The environment is partially observable, meaning agents draw observations $z \in Z$ from an observation function $O(s) : \mathcal{S} \to Z$. We denote an action-observation history by h_t, with $\mathcal{H}_t \equiv (Z \times \mathcal{A})^t$ the space of conceivable t-length histories. A ***policy*** or ***pure strategy*** π_i for player i is defined as a mapping from h_t to a probability distribution over legal actions $\pi_i : \mathcal{H} \to \Delta(\mathcal{A})$. We omit subscript i to refer to a ***joint policy*** (or ***strategy profile***) $\pi = (\pi_1, \ldots, \pi_n)$. A ***trajectory*** τ describes a rollout of game play and is represented by a sequence of tuples $\tau = \big((s_t, o_{i,t}, a_{i,t}, r_{i,t}, i)\big)_{t=1}^{|\tau|}$, where t indexes the timestep. A dataset $\mathcal{D}$ is a collection of trajectories $\{\tau^j\}_{j=1}^{|\mathcal{D}|}$, indexed here by j. We may subscript a trajectory τ_π^j to indicate the joint policy π generating the rollout. The joint probability of being in state s and executing action a under joint policy π is denoted by $\mu^\pi(s, a)$.

Multiagent interactions may also be expressed in ***normal form***. A normal form ***symmetric game*** Γ is represented by a tuple (Π, U, n), where n is the number of players, $\Pi = \Pi_1 \times \cdots \times \Pi_n$ is a set of joint strategies ($\pi \in \Pi$), and ***payoff function*** $U : \Pi \to \mathbb{R}^n$ maps strategy profiles π to vectors of payoffs (i.e., returns). A probability distribution $\sigma_i \in \Delta(\Pi_i)$ over pure strategies is called a ***mixed strategy***. We also use $\sigma_{i,s}$ to indicate player i's strategy applied at step s. A subscript $-i$ indicates all players except i. We write player i's payoff for playing π_i when the others play π_{-i} as $u_i(\pi_i, \pi_{-i})$, where u is a length-n utility vector. For mixed strategies, the payoffs are given by expectation:

$$u_i(\pi_i, \sigma_{-i}) = \mathbb{E}_{\pi_{-i} \sim \sigma_{-i}} u_i(\pi_i, \pi_{-i}),$$
$$u_i(\sigma) = \mathbb{E}_{\pi_i \sim \sigma_i} u_i(\pi_i, \sigma_{-i}).$$

Player i's ***best response*** to all others playing σ is given by $BR_i(\sigma) = \arg\max_{\sigma_i^* \in \Delta(\Pi_i)} u_i(\sigma_i^*, \sigma_{-i})$. Mixed strategy σ^* is a ***Nash equilibrium*** (NE) iff $\forall i.\ \sigma_i^* \in BR_i(\sigma^*)$. The potential gain to deviating from σ is called ***regret***: $Regret_i(\sigma) = u_i(\pi_i^*, \sigma_{-i}) - u_i(\sigma)$, where $\pi_i^* \in BR_i(\sigma)$. By definition, if σ^* is a NE, its regret is zero. An ***empirical game model*** $\hat{\Gamma} = (\hat{\Pi}, \hat{U}, n)$ is a normal-form game over a finite joint-strategy subspace $\hat{\Pi} = \hat{\Pi}_1 \times \cdots \times \hat{\Pi}_n$, $\hat{\Pi}_i \subset \Pi_i$, with payoff function $\hat{U}$ estimated through simulation.

4 CONSERVATISM IN OFFLINE GAME-SOLVING

Empirical game updates and best-response training in PSRO require explicit access to a game simulator. The offline setting, in contrast, works with fixed datasets of unspecified quality. Assessing candidate solutions requires reliable information on the included strategies *and* relevant deviations, which is contingent on the dataset's behavior coverage, or lack thereof. Since we cannot draw assumptions on the extent of this coverage, offline game-solving must account for uncertainty in regret assessments. A joint strategy π is considered ***covered*** if the dataset contains all state-action pairs visitable under π: $\forall s, a.\ \mu^\pi(s, a) > 0 \implies (s, a) \in \mathcal{D}$. In large games, full coverage of any policy π and its unilateral deviations is infeasible. Therefore, we consider the *extent* to which a policy π is covered. Our approach frames offline game-solving as an equilibrium selection problem: among solutions that seem to have low regret offline, we prefer those more certain to have low regret in the true game. By skewing offline strategy exploration toward high-confidence policy spaces, we seek to produce lower-regret solutions than alternative exploration methods with some consistency.

4.1 Training the Ensemble Dynamics Model

Our model-based approach adapts analogous single-agent offline RL approaches. We train an ensemble of K multi-layer perceptrons as dynamics models, each estimating the transition function $\mathcal{T}$, reward function $\mathcal{R}$, observation mapping O, and state-dependent action space $\mathcal{A}$, from which dataset $\mathcal{D}$ was generated. Henceforth, our references to these MDP components denote predictions of a learned dynamics model. Ensemble predictions are the average prediction across all K networks: $\overline{\mathcal{R}}_i(s_t, a_t) = \frac{1}{K} \sum_{j=1}^{K} \mathcal{R}_i^j(s_t, a_t)$, where $\mathcal{R}_i^j(s_t, a_t)$ refers to model j's prediction of player i's reward given state-action pair (s_t, a_t), and so forth for other MDP components. Additional details, including terminal handling, information state generation, and training parameters are provided in Supp. 4.1.

Prediction differences between intra-ensemble models can serve as proxy for uncertainty quantification [7]. We follow Yu et al. [32] and define ρ based on the range of model predictions:[1]

$$\rho(s_t, a_t, \mathcal{R}) = \max_{j,k} \sum_{i=1}^{n} \left| \mathcal{R}_i^j(s_t, a_t) - \mathcal{R}_i^k(s_t, a_t) \right|. \tag{1}$$

[1]The original approach of Yu et al. [32] used transition function differences. We instead employ reward differences, which achieved stronger results in our experiments.

Equation 1 reflects an assumption that models tend to provide different reward predictions on information-deprived transitions, and similar predictions for sufficiently informed transitions (whether informed explicitly or implicitly by dataset $\mathcal{D}$). The maximum prediction difference is a conservative indicator [7, 32].

4.2 A Conservative Response Objective

At each iteration of online PSRO, strategies are trained by optimizing a single agent's expected return against a fixed, stationary response target, σ_{-i}.

$$\pi_i = \arg\max_{\pi_\theta} J_i(\theta, \sigma_{-i}),$$

$$J_i(\theta, \sigma_{-i}) = \mathbb{E}_{a_t \sim (\pi_\theta, \sigma_{-i})} \sum_{t=1}^{T} \overline{\mathcal{R}}_i(s_t, a_t). \tag{2}$$

Under a fixed dataset, we cannot assume availability of covered best-responses to σ_{-i}. COffeE-PSRO modifies PSRO's response objective by optimizing an augmented reward function, $J_i'(\theta, \sigma_{-i})$

$$= \mathbb{E}_{a_t \sim (\pi_\theta, \sigma_{-i})} \left[\sum_{t=1}^{T} \overline{\mathcal{R}}_i(s_t, a_t) - \lambda \rho(s_t, a_t, \mathcal{R}) \right], \tag{3}$$

where λ is a hyperparameter. Equation 3 incorporates model-based uncertainty given σ_{-i}. Although this objective is sufficient in single-agent RL problems, offline game-solving requires information on strategy deviations to additionally *verify* equilibrium properties [4]. We hypothesize that skewing strategy exploration toward equilibria with low ρ on *unilateral deviations* will also improve performance. Suppose player i is creating a strategy exclusively optimizing for unilateral concentration. This entails creating a strategy π_i that provides other players the least uncertainty ρ on all potential responses to π_i. Fig. 1 demonstrates we can achieve this by sampling a different other-player policy at each training episode and minimizing ρ. A roughly equivalent formulation[2] is to respond to a uniform-random action policy exclusively optimizing ρ:

$$J_i''(\theta, \sigma_{-i}) = (1-\alpha) J_i'(\theta, \sigma_{-i}) - \alpha \mathbb{E}_{a_t \sim (\pi_\theta, \mathcal{U}(\mathcal{A}))} \sum_{t=1}^{T} \lambda \rho(s_t, a_t, \mathcal{R}), \tag{4}$$

where α is a hyperparameter denoting the probability of optimizing strategic deviation coverage in each training episode. In practice, we linearly anneal α to zero over PSRO iterations. In the context of strategy generation, this objective balances model-predicted reward, uncertainty given the response target σ_{-i}, and uncertainty of potential other-player deviations. Under the lens of equilibrium selection, Equation 4 can be viewed as skewing strategy exploration toward higher-certainty policy regions [28].

4.3 A Conservative Meta-Strategy Solver

We propose a novel MSS that uses ensemble model predictions to derive a conservative response target. This MSS takes concepts from the ***Bellman-consistent equilibrium learning*** (BCEL) method of Zhang et al. [33], incorporated in a game-solver based on ***replicator dynamics*** (RD) [23]. BCEL employs lower and upper utility bounds for given joint strategies and identifies the solution minimizing worst-case regret. We design an RD update that computes a stable point in heuristically estimated, worst-case regret. We define the utility predicted by model j for player i under joint policy π as $u_i^j(\pi) = \frac{1}{N} \sum_{k=1}^{N} \sum_{(s_t,a_t) \in \tau_\pi^k} \mathcal{R}_i^j(s_t, a_t)$, where returns are averaged over N trajectories τ_π^k. Lower and upper utility bounds are replaced by the minimum and maximum utility estimates over the ensemble $\mathcal{R}$, respectively: $\overline{u}_i(\pi) = \max_{j \in \{1 \dots K\}} u_i^j(\pi)$. Pessimistic utilities $\underline{u}(\pi)$ are minimized over K models. We modify the RD update by myopically optimizing for the current pessimistic regret estimate at each step:

$$UBDP \doteq \overline{u}_i(\pi_i^k, \sigma_{-i}) - \underline{u}_i(\sigma)$$

$$UBDR \doteq \max_{\pi_i^j \in (\hat{\Pi}_i \setminus \pi_i^k)} \overline{u}(\pi_i^j, \sigma_{-i}) - \underline{u}_i(\pi_i^k, \sigma_{-i})$$

$$\Delta\sigma_i^k = \sigma_i^k [UBDP - UBDR]$$

where σ_i^k is the probability player i plays strategy k and Δ indicates the amount of change applied to a variable. The first term captures an upper-bound deviation payoff (UBDP) for playing π_i^k against σ_{-i}. The second represents an upper-bound deviating regret (UBDR) if the player deviates to π_i^k. Our approach accounts for robust offline evaluation, giving our MSS its name: ***Robust Replicator Dynamics*** (R2D).

Neither RD nor R2D offer theoretical convergence guarantees. Nevertheless, in practice, RD reliably reaches exact or approximate equilibria in empirical games [29] with demonstrated success within the PSRO framework. Analogously, we demonstrate R2D's empirical benefits within COffeE-PSRO. A complementary RD variant, Regularized RD (RRD) [27], employs an early-stop condition based on a regret threshold to avoid overfitting to the current empirical game solution. R2D can analogously adopt a stopping threshold measured against worst-case regret, resulting in another MSS called ***Robust Regularized Replicator Dynamics*** (R3D).

4.4 Conservative Offline Exploration in PSRO (COffeE-PSRO)

Algorithm 1 summarizes COffeE-PSRO, depicted schematically in Fig. 1. COffeE-PSRO follows the basic structure of PSRO, using a trained dynamics model in place of the online simulator, and using Equation 4 for strategy generation. Any meta-strategy solver $\mathcal{M}$ can be used with COffeE-PSRO,[3] but our experiments suggest that R2D yields the best results.

5 EXPERIMENTS

5.1 Bargaining

We test our approach by adapting a 2-player, turn-based, imperfect information, symmetric bargaining game [13]. The game (which we refer to as *Bargaining*) features a pool of items that players must divide between them. Each player has a randomly drawn, private valuation that gives their per-unit utility value for each item. Players alternate making offers of division until one accepts

[2] Equivalence requires that the uniform distribution over strategies be behaviorally equivalent to the uniform-random strategy, which does not generally hold. In N-player games, only one player can use uniform-random action at any given episode.

[3] For conventional MSSs, empirical game payoffs would be estimated using average reward predictions rather than bounds.

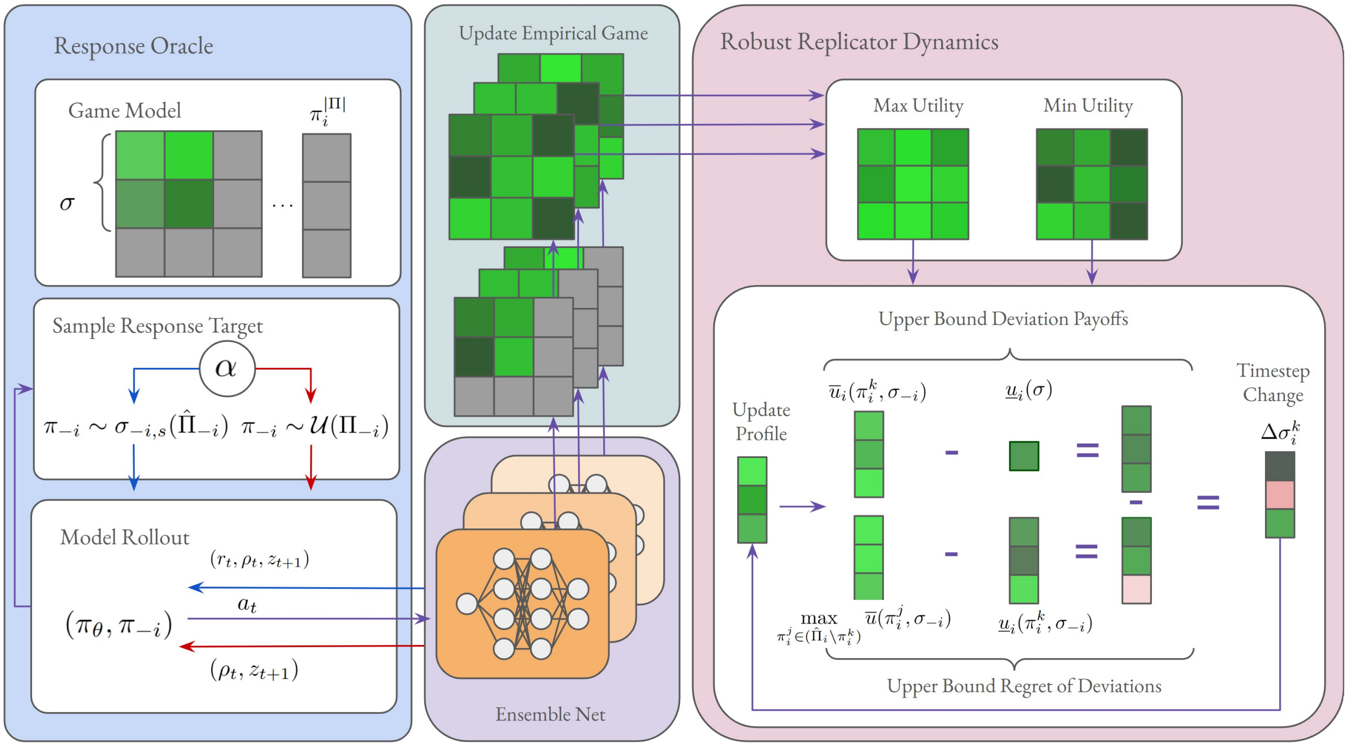


Figure 1: An iteration of COffeE-PSRO. During strategy generation, a rollout optimizes ρ against a uniformly sampled policy with probability α, or the conservative response to the target (Equation 3) otherwise. The two cases are indicated by red and blue arrows. The newly trained policy is added to the strategy set, updating the empirical game using pessimistic and optimistic utility estimates. R2D determines the next iteration's response target.

Algorithm 1 Conservative Offline Exploration PSRO (COffeE-PSRO)

Input: Dataset $\mathcal{D}$, Penalty weight λ, Coverage weight α_{init}, Steps anneal coverage weight S_α, meta-strategy solver $\mathcal{M}$, PSRO iterations S, Number of simulations N
Output: Player strategy sets $\hat{\Pi}$, Player profiles σ_S

```
Initialize empty empirical game Γ̂ : (Π̂, Û, n)
Initialize strategy sets Π̂_i ← {π_i^0}, i ∈ {1...n}
Train ensemble dynamics model of size K on D
Update Γ̂ using dynamics model with N simulations
for each PSRO iteration s from 1 to S do
    for each player i from 1 to n do
        π_i ← arg max_{π_θ} J_i''(θ, σ_{-i}) (4), using λ and α
        Update strategy sets Π̂_i ← Π̂_i ∪ {π_i}
    end for
    Update Γ̂ using dynamics model with N simulations
    Update joint profile σ_s ← M(Γ̂)
    Anneal α ← max(0, α_init(S_α − i) / S_α)
end for
return Π̂, σ_S
```

or the game reaches a maximum number of iterations. If a player accepts, players receive utility according to the accepted allocations and respective value functions, discounted by γ^t where $0 < \gamma < 1$ is a hyperparameter and t is the number of bargaining rounds so far. We set γ to .99 If no agreement is reached by the turn limit ($T = 10$), both players receive zero utility. Supp. B provides further details.

5.2 Implementation and Reproducibility

Our experiments build on an existing codebase, Deepmind's OpenSpiel [10]. We refer to datasets using tuples (N, π^β) where N is the number of trajectories and π^β is the joint behavior policy. Our experiments vary $N \in \{500, 1000, 2000\}$ and $\pi^\beta \in \{\pi^\mathcal{U}, \sigma^{eq}\}$, where $\pi^\mathcal{U}$ is a uniform-random action policy, and σ^{eq} is a uniform mixture over 5 equilibrium policies generated by online PSRO (Supp. C). Under these settings, coverage of Bargaining's full strategy space is highly unlikely, providing benchmarks that measure ability to extract approximate equilibria under limited information. All trials use Double Deep Q-Networks (DDQN) [24] for best-response calculations, yielding deterministic, pure strategies π_i. Supp. A discusses hyperparameters for our best-response and dynamics model.

We compare our approach to two benchmarks, both variants of Offline Equilibrium Finding PSRO [14]. These approaches train a dynamics model, querying it in place of an online simulator (equivalent to COffeE-PSRO with $\lambda = \alpha = 0$), and either use the output solution, or mix that solution with a behavior-cloned policy π_i^{bc} on

Table 1: Regret of final solutions, averaged over 10 trials with corresponding standard deviation and P-values, each consisting of $S = 40$ iterations. Bolded algorithmic components indicate our contributions. Datasets contain a varied number of trajectories $N \in \{500, 1000, 2000\}$ and are generated by two different behavior policies: uniform random $\pi^{\mathcal{U}}$, and a mixture over online-generated equilibria σ^{eq}. We distinguish between the MSS $\mathcal{M}_{expl}$ used during training versus $\mathcal{M}_{eval}$ during evaluation to isolate effects on strategy exploration and immediate regret [26]. All P-values are calculated with Welch's t-test to determine whether COffeE-PSRO +R2D yields significant regret differences relative to the corresponding approach. True-game regret evaluations are discussed in Supp. E.

Approach	(500, σ^{eq})	(500, $\pi^{\mathcal{U}}$)	(1000, σ^{eq})	(1000, $\pi^{\mathcal{U}}$)	(2000, σ^{eq})	(2000, $\pi^{\mathcal{U}}$)
Alg, $\mathcal{M}_{expl}$			$\mathcal{M}_{eval}$: RD			
COffeE-PSRO, R2D	**2.02 ± 0.40** | N/A	4.23 ± 0.77 | N/A	**1.10 ± 0.32** | N/A	**2.40 ± 0.55** | N/A	**0.88 ± 0.35** | N/A	**1.66 ± 0.57** | N/A
COffeE-PSRO, RD	2.58 ± 0.40 | .01	4.20 ± 0.81 | .53	1.47 ± 0.74 | .09	2.64 ± 0.55 | .17	1.64 ± 0.94 | .02	3.15 ± 2.09 | .03
OEF, **R2D**	2.71 ± 0.85 | .02	5.01 ± 1.07 | .04	1.42 ± 0.52 | .05	3.20 ± 0.37 | .01	1.03 ± 0.67 | .27	2.24 ± 0.84 | .01
OEF, RD	2.38 ± 0.55 | .06	5.09 ± 1.19 | .04	1.19 ± 0.29 | .29	3.74 ± 2.10 | .04	1.40 ± 0.79 | .04	3.28 ± 2.16 | .02
OEF-BC, **R2D**	2.91 ± 0.61 | .01	4.00 ± 0.65 | .76	2.00 ± 0.22 | .01	2.62 ± 0.22 | .13	1.85 ± 0.29 | .01	2.15 ± 0.23 | .01
OEF-BC, RD	2.82 ± 0.38 | .01	**3.85 ± 0.54** | .89	1.92 ± 0.19 | .01	2.96 ± 1.07 | .08	1.92 ± 0.49 | .01	2.83 ± 1.12 | .01
Alg, $\mathcal{M}_{expl}$			$\mathcal{M}_{eval}$: **R2D**			
COffeE-PSRO, R2D	**1.96 ± 0.45** | N/A	3.98 ± 0.60 | N/A	1.22 ± 0.41 | N/A	**2.12 ± 0.27** | N/A	**0.64 ± 0.12** | N/A	**1.40 ± 0.36** | N/A
COffeE-PSRO, RD	2.15 ± 0.43 | .17	**3.76 ± 0.75** | .76	1.34 ± 0.42 | .26	2.85 ± 0.91 | .02	1.30 ± 0.59 | .01	2.42 ± 1.43 | .05
OEF, **R2D**	2.47 ± 0.82 | .05	4.90 ± 1.11 | .02	1.08 ± 0.37 | .78	3.25 ± 1.00 | .35	0.73 ± 0.52 | .28	1.93 ± 0.49 | .01
OEF, RD	2.11 ± 0.39 | .22	4.88 ± 1.03 | .02	**1.01 ± 0.19** | .92	3.68 ± 1.63 | .01	0.96 ± 0.59 | .06	2.79 ± 1.98 | .03
OEF-BC, **R2D**	2.77 ± 0.33 | .01	3.97 ± 0.65 | .51	1.89 ± 0.16 | .01	2.70 ± 0.39 | .01	1.68 ± 0.22 | .01	2.79 ± 1.98 | .03
OEF-BC, RD	2.65 ± 0.20 | .01	3.79 ± 0.46 | .78	1.98 ± 0.19 | .01	2.92 ± 0.75 | .01	1.82 ± 0.18 | .01	2.65 ± 0.94 | .01

$\mathcal{D}$ by weight α_{bc}. We refer to the former as OEF and the latter as OEF-BC. All approaches use symmetric implementations, where players share a strategy set $\hat{\Pi}_i$ and profile σ_i, $\hat{\Gamma}$ is a symmetric game, and a single policy π_i is generated at each iteration. Analogous to how offline RL uses online simulations for return evaluation, we use online best responses (detailed in Supp. E) to evaluate true-game regret. All trials consist of $S = 40$ iterations, each iteration corresponding to one strategy generation. We tune COffeE-PSRO and OEF-BC using $\mathcal{D} = (1000, \pi^{\mathcal{U}})$.[4] COffeE-PSRO is tuned across (λ, α) for both $\mathcal{M} \in \{\text{RD}, \text{R2D}\}$, running 5 trials for each, and an additional 5 for the initially lowest regret configuration. OEF-BC is tuned exhaustively across α_{bc}, where all algorithmic variants immediately run 10 trials and then select the lowest regret parameter setting. We fix tuned parameters across datasets. Note that this tuning scheme is more exhaustive than COffeE-PSRO's. We chose to use R2D as opposed to R3D in order to isolate the effects of optimizing for conservative regret estimates over expected regret.[5] Hyperparameters are detailed in Supp. A.

5.3 Regret Analysis

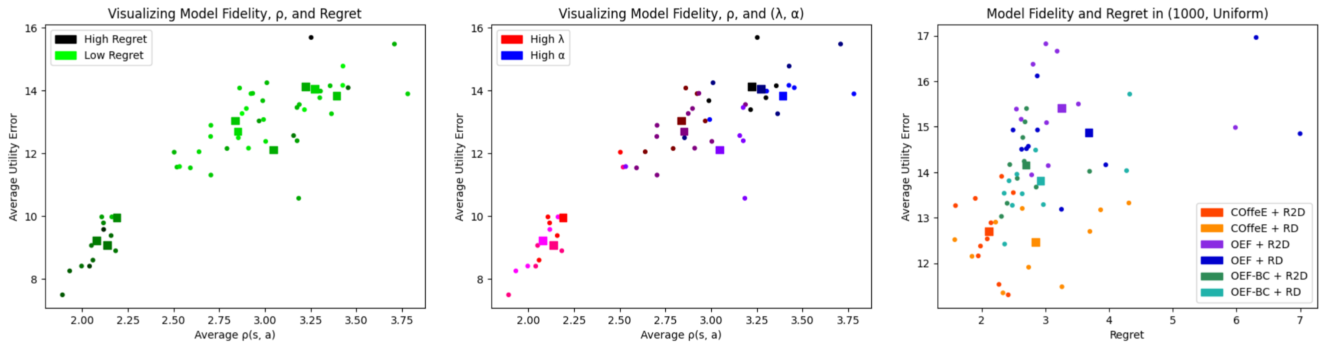
Table 1 compares tuned versions of COffeE-PSRO, OEF, and OEF-BC. We isolate and measure the improvement from an MSS used in strategy exploration $\mathcal{M}_{expl}$ by using a consistent *evaluation* MSS $\mathcal{M}_{eval}$ across different approaches [26]. Results suggest that separately applying COffeE-PSRO or R2D yields variable improvement. However, the combined use of COffeE-PSRO with R2D tends to outperform tested baselines, suggesting a strong, beneficial interplay between conservatism in strategy generation and MSS.

Comparing with OEF: Benefits of Conservatism - Except for (1000, σ^{eq}), COffeE-PSRO + R2D tends to outperform OEF, indicating that intentional conservatism and uncertainty quantification improve performance. Particularly, under a uniform policy $\pi^{\mathcal{U}}$, $\mathcal{D}$ provides relatively little information on valuable policy spaces, widening the performance margin between COffeE-PSRO + R2D and OEF. Under the same argument of conservatism, OEF-BC also outperforms OEF when using $\pi^{\mathcal{U}}$ datasets.

Comparing with OEF-BC: Strategic vs. Uninformed Conservatism - COffeE-PSRO provides consistently stronger performance than OEF-BC under all datasets except (500, $\pi^{\mathcal{U}}$). These two approaches differ conceptually in how they apply conservatism. Through strategy exploration, COffeE-PSRO operates by prioritizing search in dataset-informed spaces while OEF-BC enforces similarity to dataset behavior. These results support the notion that low-data settings provide little information on how to be conservative in a calculated manner during exploration. Thus, COffeE-PSRO's strategic conservatism is outperformed by OEF-BC's simple method of conservatism in low data, weak behavior policy regimes.

Behavior Cloning in General-Sum Games - We highlight that OEF-BC is substantially outperformed by all approaches under σ^{eq} datasets, underscoring a core property of general-sum games. Although σ^{eq} provides information on several "strong" strategies, the mixture of equilibria in general-sum games is not necessarily an equilibrium itself, making mixing with a BC-trained policy arbitrarily effective. This emphasizes a vital consideration distinguishing conservative approaches in single and multiagent environments:

[4] OEF has no additional parameters.
[5] This additionally removes the need to tune additional parameters λ_{rrd} and λ_{r3d} for fair comparison.



(a) ρ, $\delta(\hat{\Pi})$, **and regret for COffeE-PSRO tuning trials.** **(b)** ρ, $\delta(\hat{\Pi})$, **and** (λ, α) **for COffeE-PSRO tuning trials.** **(c) regret and** $\delta(\hat{\Pi})$ **for trials using** $\mathcal{D} = (1000, \pi^{\mathcal{U}})$.

Figure 2: Relationships among model fidelity, performance, and conservatism. Each datapoint is a single trial ($\mathcal{M}_{eval}$ = R2D).

imitating datasets that contain strong behavior does not imply the generation of strong strategies in the latter.

Immediate Regret Benefits of R2D - We now discuss $\mathcal{M}_{eval}$'s effect on immediately evaluated regret independent of strategy exploration. For a given approach and dataset, R2D tends to produce lower regret solutions than RD, suggesting the optimization of conservative regret estimates benefits offline equilibrium extraction. These findings are consistent with prior work [33].

5.4 Analyzing Model Fidelity

While online strategy exploration focuses on creating a set of strategies that sufficiently capture game structure, a challenge of offline game-solving is the added uncertainty of utility estimations due to limited access to data. The extent to which our algorithm should trust its offline game model for equilibrium extraction should, intuitively, be a function of its utility approximation accuracy. We refer to a game model's utility estimation accuracy as ***model fidelity*** and hypothesize that it is a key component for robust offline equilibrium discovery. More precisely, since the optimization of Equation 4 can be interpreted as skewing strategy exploration to create a higher fidelity game model, we investigate the relationship between ρ, parameter strength, model fidelity, and solution quality.

For each experimental trial depicted in Fig. 2, we use offline-generated strategies $\hat{\Pi}$ to reconstruct an empirical game $\hat{\Gamma}$ with true game simulations, calculating the average uncertainty ρ across all simulations and average utility error across all entries in $\hat{\Gamma}$: $\delta(\hat{\Pi}) = \frac{1}{\hat{\Pi}} \sum_{\pi \in \hat{\Pi}} \sum_{i=1}^{n} |u_i(\pi) - \hat{u}_i(\pi)|$, where $u_i(\pi)$ and $\hat{u}_i(\pi)$ are player i's true game and dynamics model estimated utilities, respectively. Figs. 2a and 2b present scatter plots relating ρ and $\delta(\hat{\Pi})$, color-coded proportionally to the magnitude of regret and parameters (λ, α) applied, respectively.

Plots show a clear positive relationship between ρ and $\delta(\hat{\Pi})$, indicating ρ is an effective proxy for uncertainty quantification. Parameters λ and α strongly correlate with improved model fidelity but not necessarily regret. Instead, moderate hyperparameter choices, and therefore modest model fidelity, yield the lowest regret, shown in Fig. 2b, suggesting that excessive conservatism prevents the discovery of valuable strategies even if the subgame's utility estimates are relatively accurate. These experiments demonstrate how effective offline game-solving relies on policy generation balancing conservatism and potential strategic value (model fidelity vs. regret). This tradeoff between model fidelity and regret is reflected in Fig. 2c, where the use of R2D when $\mathcal{D} = (1000, \pi^{\mathcal{U}})$ tends to decrease regret across all approaches despite compromising model fidelity.

5.5 Ablations

We provide experiments varying λ and α in Fig. 3, showing the subset of plots that ablate from the strongest parameter settings: ($\lambda = 4$, $\alpha = .2$) and ($\lambda = 4$, $\alpha = .3$) for R2D and RD, respectively. All remaining ablations of COffeE-PSRO are provided in Supp. F. Moderate levels of λ tend to produce the strongest results, both in final regret and convergence speed. However, when using $\mathcal{M}$ = R2D, a more conservative MSS, smaller values of both λ and α are favored. This indicates a threshold at which excessive conservatism from all sources becomes detrimental. Generally, higher values of α correlate to slower initial convergence, indicated by regret spikes in early iterations. These spikes are commonly followed by steep dips, sometimes finding lower regret solutions than configurations with initially stronger performance. This suggests that α, by skewing strategy exploration toward regions with higher certainty on strategic deviations, benefits lower-regret, higher-certainty equilibrium selection in later iterations.

6 LIMITATIONS

An inherent limitation of PSRO, and by extension of COffeE-PSRO, is the computational cost of training a sequence of deep RL strategies in sequence, which can be exorbitantly expensive for complex games. Several approaches have focused on reducing this computational cost [2, 22]. Our experiments find that COffeE-PSRO improves over baselines if the dataset $\mathcal{D}$ provides enough information to quantify or extrapolate uncertainty in uncovered strategy spaces (strategic conservatism). Otherwise, COffeE-PSRO provides no more benefit than uninformed conservatism. Dataset coverage is especially difficult for multiagent environments, which tend to have larger search spaces than single-agent. Lastly, experiments

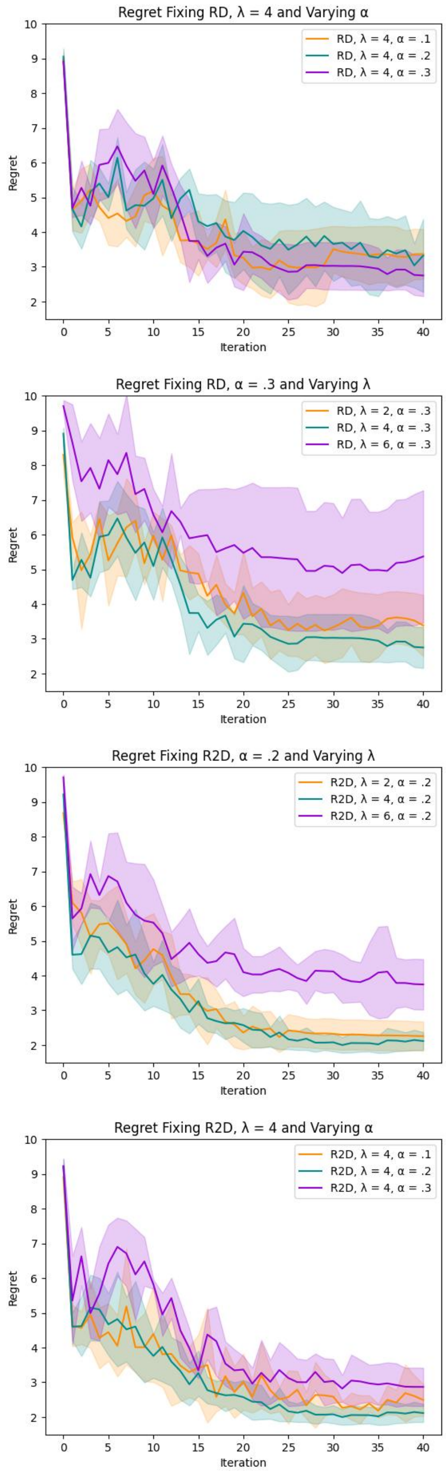


Figure 3: Regret over iterations plots under various parameter settings. Regret entries for ($\mathcal{M}$ = RD, $\lambda = 4, \alpha = .3$) and ($\mathcal{M}$ = R2D, $\lambda = 4, \alpha = .2$) are averaged over 10 trials; all others are over 5.

$\mathcal{M}$	$\lambda \downarrow$	$\alpha = .1$	$\alpha = .2$	$\alpha = .3$
RD	2	4.15 ± 1.79	3.50 ± 0.78	**3.39 ± 0.98**
	4	3.37 ± 0.80	3.33 ± 1.18	**2.58 ± 0.56**
	6	4.91 ± 0.61	**4.03 ± 1.35**	5.37 ± 2.13
R2D	2	2.89 ± 1.51	**2.26 ± 0.47**	2.50 ± 0.93
	4	2.49 ± 0.52	**2.12 ± 0.27**	2.87 ± 0.61
	6	**3.43 ± 0.55**	3.75 ± 0.81	3.64 ± 1.01

Figure 4: Final regret under various parameter settings. Regret entries for ($\mathcal{M}$ = RD, $\lambda = 4, \alpha = .3$) and ($\mathcal{M}$ = R2D, $\lambda = 4, \alpha = .2$) are averaged over 10 trials; all others are over 5. Bold and underlined numbers indicate the lowest regret for fixed λ and α, respectively. During tuning, we use the same MSS $\mathcal{M}$ for exploration and evaluation.

demonstrate the efficacy of one hyperparameter setting (λ, α) across different datasets. However, it is unclear whether these same parameters generalize to other games with different reward structures and scales.

7 CONCLUSION

We investigated a novel approach to offline game-solving, which addresses the discovery of equilibria from fixed datasets of multiagent trajectories. These datasets may provide incomplete information on relevant strategy spaces, making game-solving strictly more difficult in the offline setting. Under this added uncertainty, we framed this task as an equilibrium selection problem: of solutions apparently in approximate equilibrium, prefer those likely to minimize actual, true-game regret. We address equilibrium selection under the lens of strategy exploration, which refers to the process of deciding what strategies to include in an iteratively extended game model, by incorporating conservative algorithmic techniques from single-agent, offline RL algorithms.

Our approach, COffeE-PSRO, extends PSRO by training a dynamics model to use in place of an online simulator. Then, we design a modified response objective that optimizes model-quantified uncertainty given the current response target and potential strategy deviations. Lastly, we propose a novel MSS, R2D, that optimizes conservative regret estimates. We test our algorithm in a general-sum, sequential Bargaining game against several SOTA baselines. Our results demonstrate that COffeE-PSRO + R2D tends to produce lower regret solutions with higher consistency than SOTA approaches. We also investigated the unique relationship between algorithmic components, behavior policy, and performance in general-sum games, highlighting COffeE-PSRO's strategic method of applying conservatism. As a final solver ($\mathcal{M}_{eval}$), R2D tends to identify lower regret solutions than the RD baseline, underscoring the benefits of optimizing for pessimistic regret estimates in offline settings. Lastly, experiments analyzing model fidelity and its relationship to regret demonstrate the importance of balancing conservatism and potential strategic value in strategy exploration.

REFERENCES

[1] David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. 2019. Open-ended learning in symmetric

zero-sum games. In *36th International Conference on Machine Learning*. 434–443.
[2] Ariyan Bighashdel, Yongzhao Wang, Stephen McAleer, Rahul Savani, and Frans A. Oliehoek. 2024. Policy space response oracles: A survey. In *33rd International Joint Conference on Artificial Intelligence* (Jeju, Korea). 7951–7961.
[3] Jingxiao Chen, Weiji Xie, Weinan Zhang, Yong Yu, and Ying Wen. 2026. Offline fictitious self-play for competitive games. In *40th AAAI Conference on Artificial Intelligence* (Singapore).
[4] Qiwen Cui and Simon S. Du. 2022. When are offline two-player zero-sum Markov games solvable?. In *36th Annual Conference on Neural Information Processing Systems*. 25779–25791.
[5] Jiechuan Jiang and Zongqing Lu. 2023. Offline decentralized multi-agent reinforcement learning. In *26th European Conference on Artificial Intelligence*. 1148–1155.
[6] Patrick R. Jordan, L. Julian Schvartzman, and Michael P. Wellman. 2010. Strategy Exploration in Empirical Games. In *9th International Conference on Autonomous Agents and Multiagent Systems* (Toronto). 1131–1138.
[7] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. 2020. Morel: Model-based offline reinforcement learning. In *34th Annual Conference on Neural Information Processing Systems*. 21810–21823.
[8] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. 2019. Stabilizing off-policy Q-learning via bootstrapping error reduction. In *33rd Annual Conference on Neural Information Processing Systems*.
[9] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative Q-learning for offline reinforcement learning. In *34th Annual Conference on Neural Information Processing Systems*. 1179–1191.
[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* (2019).
[11] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *31st Annual Conference on Neural Information Processing Systems*. 4190–4203.
[12] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).
[13] Mike Lewis, Denis Yarats, Yann N. Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? End-to-end learning for negotiation dialogues. In *Conference on Empirical Methods in Natural Language Processing*. 2443–2453.
[14] Shuxin Li, Xinrun Wang, Youzhi Zhang, Jakub Cerny, Pengdeng Li, Hau Chan, and Bo An. 2022. Offline equilibrium finding. *arXiv preprint arXiv:2207.05285* (2022).
[15] Luke Marris, Paul Muller, Marc Lanctot, Karl Tuyls, and Thore Graepel. 2021. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. In *38th International Conference on Machine Learning*. 7480–7491.
[16] Paul Muller, Shayegun Omidshafiei, Mark Rowland, Karl Tuyls, Julien Pérolat, Siqi Liu, Daniel Hennes, Luke Marris, Marc Lanctot, Edward Hughes, et al. 2020. A generalized training approach for multiagent learning. In *8th International Conference on Learning Representations*.
[17] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. 2020. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359* (2020).
[18] Austin A. Nguyen, Anri Gu, and Michael P. Wellman. 2025. Explicit exploration for high-welfare equilibria in game-theoretic multiagent reinforcement learning. In *42nd International Conference on Machine Learning* (Vancouver). 45988–46007.
[19] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177* (2019).
[20] Nicolas Perez-Nieves, Yaodong Yang, Oliver Slumbers, David Henry Mguni, Ying Wen, and Jun Wang. 2021. Modelling behavioural diversity for learning in open-ended games. In *38th International Conference on Machine Learning*. 8514–8524.
[21] Zhengdao Shao, Liansheng Zhuang, Houqiang Li, and Shafei Wang. 2025. COPSRO: An Offline Empirical Game Theoretic Method With Conservative Critic. *IEEE Transactions on Neural Networks and Learning Systems* (2025).
[22] Max Olan Smith, Thomas Anthony, and Michael P. Wellman. 2021. Iterative empirical game solving via single policy best response. In *9th International Conference on Learning Representations* (Virtual).
[23] Peter D. Taylor and Leo B. Jonker. 1978. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences* 40, 1-2 (1978), 145–156.
[24] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double Q-learning. In *30th AAAI Conference on Artificial Intelligence*. 2094–2100.
[25] Xiangsen Wang, Haoran Xu, Yinan Zheng, and Xianyuan Zhan. 2024. Offline multi-agent reinforcement learning with implicit global-to-local value regularization. *Advances in Neural Information Processing Systems* 36 (2024).
[26] Yongzhao Wang, Qiurui Ma, and Michael P. Wellman. 2022. Evaluating strategy exploration in empirical game-theoretic analysis. In *21st International Conference on Autonomous Agents and Multiagent Systems*. 1346–1354.
[27] Yongzhao Wang and Michael P. Wellman. 2023. Regularization for Strategy Exploration in Empirical Game-Theoretic Analysis. In *22nd International Conference on Autonomous Agents and Multiagent Systems*. 2484–2486.
[28] Yongzhao Wang and Michael P. Wellman. 2024. Generalized response objectives for strategy exploration in empirical game-theoretic analysis. In *23rd International Conference on Autonomous Agents and Multiagent Systems*. 1892–1900.
[29] Michael P. Wellman, Karl Tuyls, and Amy Greenwald. 2025. Empirical Game-Theoretic Analysis: A Survey. *Journal of Artificial Intelligence Research* 82 (2025), 1017–1076.
[30] Yifan Wu, George Tucker, and Ofir Nachum. 2019. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361* (2019).
[31] Yaodong Yang and Jun Wang. 2021. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583v3* (2021).
[32] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. 2020. MOPO: Model-based offline policy optimization. In *34th Annual Conference on Neural Information Processing Systems*. 14129–14142.
[33] Yuheng Zhang, Yu Bai, and Nan Jiang. 2023. Offline learning in Markov games with general function approximation. In *40th International Conference on Machine Learning*. PMLR, 40804–40829.
[34] Han Zhong, Wei Xiong, Jiyuan Tan, Liwei Wang, Tong Zhang, Zhaoran Wang, and Zhuoran Yang. 2022. Pessimistic minimax value iteration: Provably efficient equilibrium learning from offline datasets. In *39th International Conference on Machine Learning*. PMLR, 27117–27142.

A HYPERPARAMETERS

Table 2: All hyperparameters

Parameter Description	Chosen Parameter
DDQN Best-Response Parameters	
Network Width	200
Network Depth	2
Replay Buffer Size	5e4
Batch Size	64
Learning Rate	1e-4
Update Target Every	1000
Learn Every	2
Discount Factor	.99
Minimum Buffer Size to Learn	5e4
Epsilon Start	1.0
Epsilon End	.02
Epsilon Linear Decay Duration	2e5
Training Steps	2e5
Dynamics Model Parameters	
Model Width	250
Model Depth	2
Ensemble Size	4
Model Training Batch Size	64
Model Training Learning Rate	3e-4
Model Training Steps	1e4
Game-Solving Parameters	
Iterations	40
Simulations per Entry	1000
Steps anneal α	10

OEF, OEF-BC, and COffeE-PSRO all used the same dynamics model parameters and architecture for fairness. Each model within an ensemble consisted of a state-transition network and a reward network, two disparate networks using the same architecture as described above. All inputs and prediction targets were normalized with the dataset's mean and standard deviation. The state-transition network was trained to predict the change in state $\Delta = s_{t+1} - s_t \mid a_t$. An observation-legal-action network was trained to share across all models within the ensemble, mapping true state to player-specific observations and legal-action masks. We train each dynamics model using mean-squared-error loss and the parameters listed in Table 2. Observation action histories are converted to the game's defined information state through an internal module. An episode is considered finished if either a rollout reaches a maximum length or the resulting state prediction is close (mean absolute difference < .5) to a predefined terminal state. Bargaining-specific details for dynamics model training are provided in Sec B. Best-response parameters for the DDQN are provided in Table 2. These parameters were not tuned but fixed at sensible values.

B BARGAINING GAME ADAPTATION

We designed a generalized version of the Bargaining [13] provided by OpenSpiel [10]. Our game is defined by several parameters, summarized in Table 3.

Table 3: Parameters in the bottom half are sampled at the beginning of each episode while those in the top half are held constant or define distributions that bottom-half parameters sample from.

Symbol	Chosen	Description
N_{items}	3	Number of item types
$[v_{min}, v_{max}]$	[5, 10]	Summed valuation bounds for both players $i \in \{1, 2\}$.
$[c_{min}, c_{max}]$	[5, 7]	Summed item count bounds for any sampled pool
T	10	Max game length
γ	.99	Discount factor of valuation
$\{c^j\}_{j=1}^{N_{items}}$	N/A	The number of items of each item type in the pool
$\{v_i^j\}_{j=1}^{N_{items}}$	N/A	Valuations for players $i \in \{1, 2\}$

At the beginning of each episode, an item pool is sampled uniformly from all possible pools that satisfy two conditions:

$$c^j \geq 1 \;\forall j \in \{1 \dots N_{items}\}$$
$$\sum_{j=1}^{N_{items}} c^j \in [c_{min}, c_{max}]$$

where c^j is a positive integer. An item pool is represented by a vector of item counts $C \doteq \{c^j\}_{j=1}^{N_{items}}$. Next, player valuations are sampled uniformly from all possible valuations that satisfy two conditions:

$$v_i^j \geq 1 \;\forall j \in \{1 \dots N_{items}\}, i \in \{1, 2\}$$
$$\sum_{i=j}^{N_{items}} v_i^j \in [v_{min}, v_{max}] \;\forall i \in \{1, 2\}$$

where v_i^j is a real number greater than or equal to 1. Player i's valuation is represented by the vector $V_i \doteq \{v_i^j\}_{j=1}^{N_{items}}$. For symmetry, the first player is decided with a coin flip and valuation sample distributions are identical for both players. The game terminates if a player accepts an offer or the number of turns exceeds T. Our chosen parameters are listed in Table 3

The state, observation, information-state (history), and action vectors are represented by

$$s_t = \{\{\mathbb{I}[a_{t-1} = ACCEPT]\} + \{t\} + C + V_1 + V_2 + \{a_{t-1}\} + \{i_{curr}\}$$
$$s_{terminal} = \{-1\}^{|s \in \mathcal{S}|}$$
$$a_t = \{n^j\}_{j=1}^{N_{items}} \mid n^j \in \{0, \dots c^j\}$$
$$o_{i,t} = \{\mathbb{I}[a_{t-1} = ACCEPT]\} + \{t\} + C + V_i + \{a_{t-1}\}$$
$$h_{i,t} = \{\mathbb{I}[a_{t-1} = ACCEPT]\} + \{t\} + C + V_i + \cup_{t'=0}^{t'-1} a_{t'}$$

where "+" denotes concatenation.

Note that the output of DDQN assumes a discrete action space represented by an index, not an integer vector a_t. Therefore, the network output size equals the number of all possible offers under all possible pools, computable given the parameters in Table 3. Furthermore, since the DDQN is not able to output vectorized actions as shown above, we create an internal module that converts action indices a_t^{index} to offer vectors a_t of length N_{items}. We also create a module that converts an explicit history $\{o_{i,t'}, a_{1,t'}, a_{2,t'}\}_{t'=1}^{t}$ into a condensed information state $h_{i,t}$.

C DATASET GENERATION

Experiments used three different dataset sizes $N \in \{500, 1000, 2000\}$ and two different behavior policies $\pi \in \{\pi^{\mathcal{U}}, \sigma^{eq}\}$, referring to each dataset by the tuple (N, π). $\pi^{\mathcal{U}}$ denotes a joint policy that selects legal actions at uniform random: $\pi^{\mathcal{U}} : \mathcal{U}(\mathcal{A})$. σ^{eq} denotes a uniform strategy mixture over 5 profiles generated trials of PSRO, each consisting of 20 iterations: $\sigma^{eq} : \mathcal{U}(\{\sigma^{PSRO_i}\}_{i=1}^{5})$, where uniform sampling occurs at the beginning of each episode. For independence, all offline experiments generated independent datasets $\mathcal{D}$ at the beginning of every trial using the listed predefined joint policies.

D OTHER ALGORITHMIC DETAILS

D.1 Policy Space Response Oracles

The PSRO algorithm [11] iteratively extends an empirical game model $\hat{\Gamma} = (\hat{\Pi}, \hat{U}, n)$ using RL. We initialize $\hat{\Pi}_i$, for example, with a singleton random policy for each player i. PSRO extends $\hat{\Gamma}$ at each iteration through these steps: (1) update $\hat{U}$ by estimating payoffs for all ***strategy profiles*** (i.e., joint strategies $\hat{\Pi}$) through simulation, (2) extract a ***target profile*** σ by applying a ***meta-strategy solver*** (MSS) to $\hat{\Gamma}$, and (3) derive a ***best-response policy*** $\pi_i^\star \in \Pi_i$ for each player i by applying deep RL in an environment where other players are fixed to play σ_{-i} and add $\pi_i^\star$ to $\hat{\Pi}_i$. PSRO is summarized in Algorithm 2.

Algorithm 2 Policy Space Response Oracles (PSRO)

Input: Meta-strategy solver $\mathcal{M}$, PSRO iterations S, Number of Simulations N
Output: Player strategy sets Π, Player Profiles σ_S

Initialize empty empirical game $\hat{\Gamma} : (\hat{\Pi}, \hat{U}, n)$
Initialize strategy set w/ random policy $\hat{\Pi}_i \leftarrow \{\pi_i^0\}, i \in \{1, \dots, n\}$
Update empirical game $\hat{\Gamma}$ with N simulations
for each PSRO iteration s to S **do**
  for each player i to n **do**
    Define $J_i(\theta) = \mathbb{E}_{\pi_{-i} \sim \sigma_{-i}} [\sum_{t=1}^{T} r_{i,t}]$
    $\pi_i^\star \approx \arg\max_\pi J_i(\theta)$
    Update strategy sets $\hat{\Pi}_i \leftarrow \hat{\Pi}_i \cup \pi_i^\star$
  end for
  Update empirical game $\hat{\Gamma}$ with N simulations
  Update joint profile $\sigma_s \leftarrow \mathcal{M}(\hat{\Gamma})$
end for
return $\hat{\Pi}, \sigma_S$

D.2 Offline Equilibrium Finding

Algorithm 3 Offline Equilibrium Finding - Behavior Cloning (OEF, OEF-BC)

Input: Dataset $\mathcal{D}$ of trajectories, meta-strategy solver $\mathcal{M}$, PSRO iterations S, Number of Simulations N, Behavior cloning mixing weight α_{bc}
Output: Player bc-mixed strategy sets Π^{bc}, Player profiles σ_S

1: Initialize empty empirical game $\hat{\Gamma} : (\hat{\Pi}, \hat{U}, n)$
2: Initialize strategy set w/ random policy $\hat{\Pi} \leftarrow \{\pi_i^0\}$
3: Train ensemble dynamics model of size K on $\mathcal{D}$
4: Train π_i^{bc} using behavior cloning on $\mathcal{D}$ for $i \in \{1, \dots n\}$
5: Update empirical game $\hat{\Gamma}$ using dynamics model with N simulations.
6: **for** each PSRO iteration s to S **do**
7:   **for** each player i to n **do**
8:     Define $J_i(\theta) = \mathbb{E}_{a_t \sim (\pi_\theta, \sigma_{-i})} [\sum_{t=1}^{T} \overline{\mathcal{R}}_i(s_t, a_t)]$
9:     $\pi_i \approx \arg\max_{\pi_\theta} J(\theta)$
10:     Update strategy sets $\hat{\Pi}_i \leftarrow \hat{\Pi}_i \cup \pi_i$
11:   **end for**
12:   Update empirical game $\hat{\Gamma}$ using dynamics model with N simulations
13:   Update joint profile $\sigma_S \leftarrow \mathcal{M}(\hat{\Gamma})$
14: **end for**
15: **for** each player i to n **do**
16:   Mix policies $\Pi_i^{bc} = \cup_{s=1}^{S} \pi_i^{s,mix}$ where $\pi_i^{s,mix}(a \mid s) = \pi_i^s(a \mid s)(1 - \alpha_{bc}) + \alpha_{bc}\pi_i^{bc}(a \mid s)$
17: **end for**
18: **return** Π_i^{bc}, σ^S

Offline equilibrium finding [14], as adapted to PSRO, is summarized in Alg. 3. The approach trains a dynamics model on dataset $\mathcal{D}$ and uses it in place of an online simulator. In our approach, we use the ensemble architecture and Double Deep Q-Learning as described in Sections 4.1 and 5.2, respectively, for a fair comparison with COffeE-PSRO. OEF also trains a behavior cloned policy π^{bc} on $\mathcal{D}$, which is used at the end of training to mix with the output policy $\pi \in \Pi$ as indicated by Line 16. Note that mixing implies sampling action a_t from π_i^{bc} or π_i^s at each timestep. Li et al. [14] chooses α_{bc} *at test time* using a true game simulator to determine which setting has the lowest exploitability. While this can be seen as a violation of the offline learning assumption, we evaluate COffeE-PSRO against OEF-BC(α_{bc}) with the strongest test-time results. The OEF baseline is identical to OEF-BC but sets α_{bc} to 0.

E TRUE GAME REGRET

Offline-generated profiles are evaluated in the true game. We train an online best response to profiles outputted at the last S_{eval} iterations: $\Pi_i^{online} = \{BR_i(\sigma_{i,s}) \mid s \in \{T - S_{eval}, \dots T\}\}$, where s indicates the s-th profile generated and T denotes the number of PSRO iterations. We construct a combined empirical game that includes offline and online trained strategies $\Pi_i^{eval} = \Pi_i \cup \Pi_i^{online}$. Then we construct a normal-form game using Π_{eval}, recomputing all entries using $N = 1000$ *true game* simulations. Player i's regret given iteration s's joint profile is approximated by the following:

$$Regret_i(\sigma_s) \approx \max_{\pi_i^{eval} \in \Pi_i^{eval}} u_i(\pi_i^{eval}, \sigma_{-i,s}) - u_i(\sigma_s)$$

All results report the sum of regret across players $\sum_{i=1}^{n} Regret_i(\sigma_s)$. Ablations use $S_{eval} = T = 40$ while all other experiments use $S_{eval} = \frac{T}{2} = 20$.

F ADDITIONAL RESULTS

F.1 Regret Results During Tuning

Table 4: True game regret values of final profiles outputted by baseline runs. Each trial consists of 40 iterations, where an iteration is one new strategy, and reported values are averaged over 10 trials. Note that when $\alpha_{bc} = 0.0$, we refer to the baseline named OEF.

α_{bc}	OEF-BC + RD	OEF-BC + R2D
0.0	3.74 ± 2.10	3.25 ± 1.00
0.2	2.96 ± 1.07	2.70 ± 0.39
0.4	3.49 ± 0.25	3.50 ± 0.22
0.6	4.79 ± 0.25	4.87 ± 0.19
0.8	6.56 ± 0.28	6.70 ± 0.19

We provide final true game regret values for tested baselines in Table 4 and all ablations over COffeE-PSRO parameters (λ, α, $\mathcal{M}$) in Table 5 and Fig. 5.

Table 5: True game regret values of final profiles outputted by COffeE-PSRO during tuning. Underlined values are averaged over 10 trials, 5 otherwise.

$\mathcal{M}$	$\lambda \downarrow$	$\alpha = .1$	$\alpha = .2$	$\alpha = .3$
RD	2	4.15 ± 1.79	3.50 ± 0.77	3.39 ± 0.98
	4	3.37 ± 0.80	3.33 ± 1.18	<u>2.64 ± 0.55</u>
	6	4.91 ± 0.61	4.03 ± 1.35	5.37 ± 2.13
R2D	2	2.89 ± 1.51	2.26 ± 0.47	2.50 ± 0.93
	4	2.49 ± 0.52	<u>2.12 ± 0.27</u>	2.87 ± 0.61
	6	3.43 ± 0.55	3.75 ± 0.81	3.64 ± 1.01

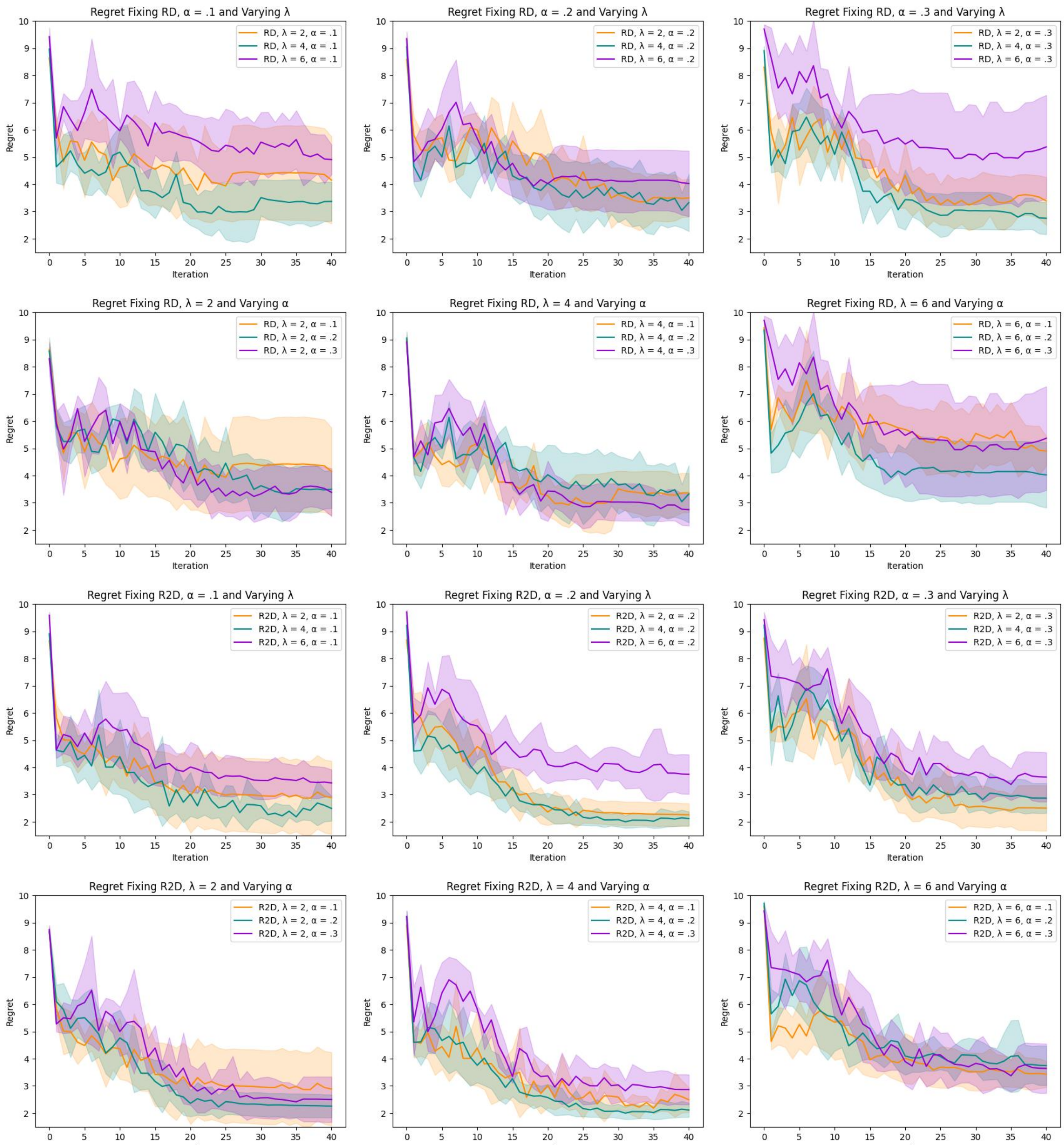


Figure 5: Ablation over COffeE-PSRO parameters. The top two rows ablate over α and λ, fixing $\mathcal{M}$ to RD, while the next two rows fix $\mathcal{M}$ to R2D.