Regularization for Strategy Exploration in Empirical Game-Theoretic Analysis

Yongzhao Wang , **Michael P. Wellman**
University of Michigan
{wangyzh, wellman}@umich.com

Abstract

In iterative approaches to empirical game-theoretic analysis (EGTA), the strategy space is expanded incrementally based on analysis of intermediate game models. A common approach to *strategy exploration*, represented by the double oracle algorithm, is to add strategies that best-respond to a current equilibrium. This approach may suffer from overfitting and other limitations, leading the developers of the policy-space response oracle (PSRO) framework for iterative EGTA to generalize the target of best response, employing what they term *meta-strategy solvers* (MSSs). Noting that many MSSs can be viewed as perturbed or approximated versions of Nash equilibrium, we adopt an explicit regularization perspective to the specification and analysis of MSSs. We propose a novel MSS called *regularized replicator dynamics* (RRD), which simply truncates the process based on a regret criterion. We show that RRD outperforms existing MSSs in various games. We extend our study to three-player games, for which the payoff matrix is cubic in the number of strategies and so exhaustively evaluating profiles may not be feasible. We employ a profile search method that can identify solutions from incomplete models, and combine this with iterative model construction using a regularized MSS. Finally, we find through experiments that the regret of best-response targets is a strong indicator for the performance of strategy exploration, which provides an explanation for the effectiveness of regularization in PSRO.

1 Introduction

The methodology of ***empirical game-theoretic analysis*** (EGTA) [Tuyls *et al.*, 2020; Wellman, 2016] provides a broad toolbox of techniques for game reasoning with models based on simulation data.[1] As many multiagent systems of interest are not easily expressed or tackled analytically, EGTA offers an alternative approach whereby a space of strategies is examined through simulation, combined with game model induction and inference. The number of strategies that can be explicitly incorporated in game models is significantly limited by computational constraints, hence the selection of strategies to include is pivotally important. For accurate analysis results, we require that the included strategies are high-performing and cover the key strategic issues [Balduzzi *et al.*, 2019]. The challenge of efficiently assembling an effective portfolio of strategies for EGTA is called the ***strategy exploration*** problem [Jordan *et al.*, 2010].

Strategy exploration in EGTA is most clearly formulated within an iterative procedure, whereby generation of new strategies is interleaved with game model estimation and analysis. The ***Policy Space Response Oracle*** (PSRO) algorithm of Lanctot *et al.* [2017] provides a flexible framework for iterative EGTA, where at each iteration, new strategies are generated through reinforcement learning (RL). The learning player trains in an environment where other players are fixed in a profile (pure or mixed) comprising strategies from previous iterations. The key design question is how to set the other-player profile to be employed as a training target. In PSRO, the component that derives this target is called a ***meta-strategy solver*** (MSS), as it takes an empirical game model as input and "solves" it to produce the target profile. The learning agent then employs RL to search for a strategy best-responding to the MSS target. In effect, specifying an MSS defines the strategy exploration method for PSRO.

An obvious choice for MSS is the solution concept employed as the objective game analysis, typically Nash equilibrium (NE). Incrementally adding strategies that are best-responses to NE of the current strategy set is known as the ***double oracle*** (DO) algorithm [McMahan *et al.*, 2003], and PSRO with NE as MSS is essentially DO with RL for computing (approximate) best response. Though DO is often effective, there is ample evidence that best-response to NE is not always the best approach to strategy exploration. Schvartzman and Wellman [2009a] observed cases where it would approach a true equilibrium extremely slowly, such that even adding random strategies could provide substantial speedups. More generally, Lanctot *et al.* [2017] argued that best-responding to Nash overfits to the current equilibrium strategies, and thus tends to produce results that do not generalize to the overall space. This was indeed their major motivation for defining a generalized MSS concept for strategy exploration. For example, as an alternative MSS Lanctot *et al.*

[1] A table of acronyms is provided in Appendix A.

[2017] proposed *projected replicator dynamics* (PRD), which employs a replicator dynamics (RD) search for equilibrium, truncating the replicator updates to ensure a lower bound on probability of playing each pure strategy.

We take a further step in this direction and adopt an explicit regularization perspective to the specification and analysis of MSSs. We propose a novel MSS called ***regularized replicator dynamics*** (RRD), which truncates the NE search process in intermediate game models based on a regret criterion. Specifically, at each iteration of PSRO, the best-response target profile is updated by running RD, stopping if the regret of the current profile with respect to the empirical game meets a specified regret threshold. The regret threshold is a hyperparameter, which may be adjusted to suit a particular game class, or annealed to control the degree of regularization across iterations. We assess the performance of RRD in various games and show that RRD outperforms several existing MSSs in terms of convergence rate and quality of intermediate empirical game models.

As the size of a payoff matrix is exponential in the number of players, the cost of maintaining completely specified models over the iterations of PSRO can be prohibitive beyond two players. To mitigate this issue, we employ a PSRO-compatible profile search method, called ***backward profile search*** (BPS), which finds solution concepts without simulating the whole payoff matrix. We combine RRD with BPS, and demonstrate the effectiveness of this combination in a three-player game.

Finally, our experiments shed light on the source of the benefit of regularization for strategy exploration. Across a variety of settings, we find that the approximate empirical-game NE produced by RRD tend to have *lower regret in the full game*, compared to exact NE of the empirical game. This not only provides an explanation for the benefits of regulation, it may also suggest a way to evaluate the potential of novel MSS designs in PSRO-related approaches.

Contributions of this study include:

1. RRD: a novel MSS that truncates the NE search process in intermediate game models based on a regret criterion. Our MSS exhibits desired proprieties for strategy exploration (e.g., improved adaptability across games) compared to previous MSSs. We demonstrate that RRD outperforms relevant alternatives from the literature in various games.

2. A comprehensive analysis of learning with RRD, including the performance stability with respect to the regret threshold.

3. Integration of method for selective profile evaluation with PSRO, enabling game-solving without simulating the whole payoff matrix. We combine RRD with this method and show its effectiveness for learning in a three-player game.

4. Demonstration of the key relationship between regret of best-response targets and the performance of MSSs in PSRO.

2 Related Work on Strategy Exploration

The first instance of automated strategy generation in EGTA was a genetic search over a parametric strategy space, optimizing performance against an equilibrium of the empirical game [Phelps *et al.*, 2006]. Schvartzman and Wellman [2009b] deployed tabular RL as a best-response oracle in EGTA for strategy generation. These same authors framed the general problem of ***strategy exploration*** in EGTA and investigated whether better options exist beyond best-responding to an equilibrium [Schvartzman and Wellman, 2009a]. Jordan *et al.* [2010] further extended this line of work by adding strategies that maximize the deviation gain from an empirical rational closure.

Investigation of strategy exploration was advanced significantly by introduction of the PSRO framework [Lanctot *et al.*, 2017]. PSRO applied deep RL as an approximate best-response oracle to certain designated other-agent profile selected by the MSS. When employing NE as MSS, PSRO reduces to the DO algorithm [McMahan *et al.*, 2003]. To generate strategy effectively, Lanctot *et al.* [2017] balanced between overfitting to NE and generalizing to the strategy space outside the empirical game, and proposed ***projected replicator dynamics*** (PRD), which employs an RD search for equilibrium [Taylor and Jonker, 1978; Smith and Price, 1973] and ensures a lower bound on probability of playing each pure strategy. For simplicity, we often refer to an MSS as shorthand for PSRO with that MSS when the context is unambiguous. For example, we may say "PRD" to mean "PSRO with PRD".

PSRO can also be viewed as generalizing some classic game-learning dynamics. For example, selecting a uniform distribution over current strategies as MSS essentially reproduces the classic ***fictitious play*** (FP) algorithm [Brown, 1951]. Moreover, an MSS that simply extracts the most recent strategy duplicates the ***iterated best response*** algorithm. Note that the MSSs generating these dynamics are solvers in only a trivial sense; they do not substantively employ an empirical game model as they derive from the strategy sets directly.

Following the line of PSRO, some works propose MSSs that effectively *regularize* the target profile to prevent from best-responding to an exact equilibrium. Specifically, Wang *et al.* [2019] employed a mixture of NE and uniform, which essentially samples whether to apply DO or FP for every PSRO iteration, thus illustrating the possibility of combining MSSs. Wright *et al.* [2019] added an adjustment step to DO, which fine-tunes the generated policy network against a mix of previous equilibrium strategies. Balduzzi *et al.* [2019] introduced a new MSS, called ***rectified Nash***, designed to increase diversity of empirical strategy space. Dinh *et al.* [2022] proposed an MSS for two-player zero-sum game that applies online learning to the empirical game and outputs the online profile as a best-response target. Beyond selecting NE as a solution concept, Muller *et al.* [2020] proposed a new MSS based on an evolutionary-based concept, α-rank [Omidshafiei *et al.*, 2019], and Marris *et al.* [2021] proposed maximum welfare coarse correlated equilibrium (MWCCE), and maximum Gini coarse correlated equilibrium (MGCCE) for computing correlated equilibria, within the PSRO framework.

3 Preliminaries

A normal-form game $\mathcal{G} = (N, (S_i), (u_i))$ comprises a finite set of players N indexed by i, a non-empty set of strategies S_i for player $i \in N$, and a utility function $u_i : \prod_{j \in N} S_j \to \mathbb{R}$ for player $i \in N$.

A mixed strategy σ_i is a probability distribution over strategies in S_i, with $\sigma_i(s_i)$ denoting the probability player i plays strategy s_i. We adopt conventional notation for the other-agent profile: $\sigma_{-i} = \prod_{j \neq i} \sigma_j$. Let $\Delta(\cdot)$ represent the probability simplex over a set. The mixed strategy space for player i is given by $\Delta(S_i)$. Similarly, $\Delta(S) = \prod_{i \in N} \Delta(S_i)$ is the mixed profile space.

Player i's ***best response*** to profile σ comprises strategies yielding maximum payoff for i, fixing other-player strategies:

$$br_i(\sigma_{-i}) \equiv \operatorname*{argmax}_{\sigma'_i \in \Delta(S_i)} u_i(\sigma'_i, \sigma_{-i}).$$

Let $br(\sigma) \equiv \prod_{i \in N} br_i(\sigma_{-i})$ be the overall best-response correspondence for a profile σ. A ***Nash equilibrium*** (NE) is a profile σ^* such that $\sigma^* \in br(\sigma^*)$.

Player i's ***regret*** for profile σ in game $\mathcal{G}$ is given by

$$\rho_i^{\mathcal{G}}(\sigma) \equiv \max_{s'_i \in S_i} u_i(s'_i, \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}).$$

Regret captures the maximum player i can gain in expectation by unilaterally deviating from its mixed strategy in σ to an alternative strategy in S_i. An NE has zero regret for every player. A profile is said to be an ***ϵ-Nash equilibrium*** (ϵ-NE) if no player can gain more than ϵ by unilateral deviation. We define the regret of a strategy profile σ as the sum over player regrets:

$$\rho^{\mathcal{G}}(\sigma) \equiv \sum_{i \in N} \rho_i^{\mathcal{G}}(\sigma).$$

A ***restricted game*** $\mathcal{G}_{S \downarrow X}$ is a projection of full game $\mathcal{G}$, in which players choose from restricted strategy sets $X_i \subseteq S_i$. An ***empirical game*** $\hat{\mathcal{G}}$ is a model of true game $\mathcal{G}$ where payoffs are estimated through simulation. Thus, $\hat{\mathcal{G}}_{S \downarrow X} = (N, (X_i), (\hat{u}_i))$ denotes an empirical game model where $\hat{u}$ is an estimated projection of u onto the strategy space X.

Replicator dynamics (RD) describes an evolving trajectory of mixed profiles, inspired by natural selection [Taylor and Jonker, 1978; Smith and Price, 1973]. RD is commonly employed as a heuristic equilibrium search algorithm. We consider a discrete form of RD, where player i's probability of playing strategy is updated in proportion to its payoff for deviating to that strategy from the current mixture. Mathematically, the replicator equation for player i's strategy s_i in a current profile σ is given by

$$\frac{d\sigma_i(s_i)}{dt} = \sigma_i(s_i)[u_i(s_i, \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i})].$$

At each iteration of RD, player i's mixed strategy σ_i is updated by $\sigma_i \leftarrow P(\sigma_i + \alpha \frac{d\sigma_i}{dt})$, where α is a step size for RD and P is a projection operator to the strategy simplex, namely $P(\sigma_i) = \operatorname{argmin}_{\sigma'_i \in \Delta} \|\sigma'_i - \sigma_i\|_2$.

PSRO is presented below as Algorithm 1. Choice of MSS dictates the strategy exploration approach.

Algorithm 1 PSRO, parametrized by solver MSS

```
Input: initial strategy sets $X$
Estimate $\hat{\mathcal{G}}_{S\downarrow X}$ by simulating $\sigma \in X$
Initialize target $\sigma_i \leftarrow \text{Uniform}(X_i)$
for PSRO iteration $\tau = 1, 2, \ldots, \mathcal{T}$ do
    for player $i \in N$ do
        for many RL training episodes do
            Sample a profile $s_{-i} \in \sigma_{-i}$
            Train BR oracle $s'_i$ against $s_{-i}$
        end for
        $X_i \leftarrow X_i \cup \{s'_i\}$
    end for
    Update $\hat{\mathcal{G}}_{S\downarrow X}$ by simulating missing profiles over $X$
    Compute best-response target $\sigma \leftarrow \text{MSS}(\hat{\mathcal{G}}_{S\downarrow X})$
end for
Return $\hat{\mathcal{G}}_{S\downarrow X}$
```

4 Regularization for Strategy Exploration

4.1 Regularized Replicator Dynamics

To avoid overfitting a response to NE, we adopt an explicit regularization perspective on strategy exploration. Specifically, we propose a method to derive approximate NE by truncating an RD-based search. Our new MSS, called ***regularized RD*** (RRD), simply runs RD on the empirical game, stopping when the regret of the current profile (w.r.t the empirical game) meets a specified regret threshold λ, or a maximum number of iterations is reached. In the RRD procedure (Algorithm 2), each player's strategy is initialized with a uniform distribution over strategies in the empirical game. Then the replicator equation is iteratively applied until the regret of the current profile (w.r.t the empirical game) becomes smaller than the regret threshold λ. Since RD does not generally converge to an exact equilibrium, there is no guarantee a finite regret threshold λ will ever be reached. We therefore set a maximum number of iterations M, and if the limit is reached return the profile with the lowest regret found to that point.

Note that RRD supports direct control of the degree of regularization through an explicit parameter: the regret threshold. This parameter is meaningful across games with different strategy sets, as long as the utility scales on which regret is measured are comparable.

Algorithm 2 RRD

```
Parameters: regret threshold $\lambda$, RD step size $\alpha$
Input: an empirical game $\hat{\mathcal{G}}_{S\downarrow X}$
Initialize RD with $\sigma_i \leftarrow \text{Uniform}(X_i)$
while $\rho^{\hat{\mathcal{G}}_{S\downarrow X}}(\sigma) > \lambda$ do
    for player $i \in N$ do
        $\sigma_i \leftarrow P(\sigma_i + \alpha \frac{d\sigma_i}{dt})$
    end for
end while
Return $\sigma$
```

The procedure of PSRO with RRD is obtained by employing RRD as the MSS in Algorithm 1. A slight modification readily supports annealing schemes where the regret threshold for RRD is varied across PSRO iterations.

4.2 Selective Profile Evaluation using BPS

One obstacle to scaling PSRO is that the size of the empirical game grows exponentially in the number of players. Even in games with only a few players (e.g., three or four), exhaustive simulation of the payoff matrix may become infeasible as the strategy space grows. Fortunately, it is often possible to derive solutions from only partial payoff matrices [Fearnley *et al.*, 2015], and prior EGTA researchers have developed methods for selectively evaluating strategy profiles in search for equilibrium [Jordan *et al.*, 2008; Sureka and Wurman, 2005]. For example, the algorithm of Brinkman and Wellman [2016] maintains a set of ***complete subgames*** of the empirical game (i.e., maximal strategy sets over which profiles have been exhaustively evaluated). NE of these subgames are proposed as candidate equilibria of the empirical game. For each candidate, all of the one-player deviations to strategies outside the subgame are evaluated. If there is no beneficial deviation to the candidate, then the candidate is ***confirmed*** as an NE of the empirical game. Otherwise, subgames are extended by the beneficially deviating strategies and the search continues.

We developed a simple profile search method for PSRO, which we call ***backward profile search*** (BPS, Algorithm 3). Our method resembles that of Brinkman and Wellman [2016], but takes into account the sequence in which the strategies were generated. At each iteration, BPS starts search from the strategies most recently added to the empirical game by PSRO, then searches potential deviations backward across previous PSRO iterations. Our motivation is that the newest strategies are most likely to participate in equilibria. Once BPS confirms a solution of the empirical game, we apply RRD to the subgame over the support of this solution. By construction, this subgame is completely evaluated (as required by RD), whereas the entire empirical game payoff matrix is only partially evaluated. In our experiments, we show that BPS can successfully find best-response targets in a three-player game, short of exhaustive evaluation of the empirical game.

Figure 1 illustrates the mechanism of BPS at the third iteration of PSRO, in which each player has four strategies. The $4 \times 4 \times 4$ cube in Figure 1 represents the payoff matrix of the current empirical game. Green cells represent the payoffs of profiles that have been evaluated from previous iterations and white cells represent the payoffs of potential deviations from the equilibrium of the current subgame. The missing cells represent payoffs of profiles that have not been evaluated. Note that the current payoff matrix is incomplete. To find the NE of the empirical game, BPS starts search from evaluating the subgame constituted by the most recently added strategy of each player, represented by the red cell. Since the red cell is a pure-strategy profile, it is also the NE of the current subgame. Next, BPS evaluates the payoffs of all potential deviations (white cells) from the red cell. Suppose the blue cell is a profile with the largest deviation payoff for certain player i. Then player i adds the corresponding deviation strategy to her strategy set of the current subgame. Now the profile space of the current subgame contains the red and the blue profiles. Then BPS repeats evaluating all profiles in the current subgame, computing NE of the current subgame, and evaluating potential deviations from the NE. Again, if the purple cell is a deviation profile, then the corresponding deviating strategy will be added to the strategy set of the subgame. BPS repeats this procedure until the NE of the subgame is confirmed, that is, no beneficial deviation could be found in the empirical game.

Algorithm 3 Backward Profile Search

```
Input: Empirical game $\hat{\mathcal{G}}_{S\downarrow X}$ with partial payoff matrix.
Initialize subgame with strategy sets $Z = (Z_i)$, $i \in N$,
    where $Z_i = \{s^i_\tau\}$, with $s^i_\tau$ the player $i$ strategy added in
    the most recent PSRO iteration $\tau$.
while True do
    $\sigma \leftarrow$ NE_search($\hat{G}_{S\downarrow Z}$)
    deviation_exists $\leftarrow$ False
    for player $i \in N$ do
        $s_i \leftarrow \mathrm{argmax}_{s' \in X_i}\, u_i(s', \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i})$
        if $s_i \notin Z_i$ then
            $Z_i \leftarrow Z_i \cup \{s_i\}$
            deviation_exists $\leftarrow$ True
        end if
    end for
    if $\neg$ deviation_exist then
        return $\sigma$
    end if
    Evaluate missing profiles of $Z$ through simulation.
end while
```

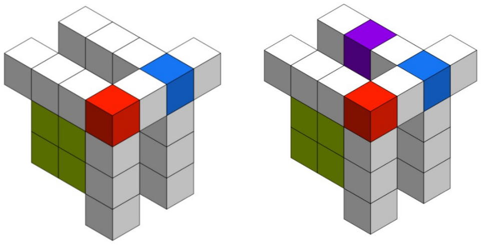


Figure 1 An illustration of a partial payoff matrix of a three-player empirical game and the workflow of BPS. Green: evaluated profiles from previous PSRO iterations; White: deviation profiles from NE of current subgame; Red: the profile with the most recently added strategies; Blue and purple: profiles with the largest deviation payoffs.

5 Performance and Analysis

5.1 Experimental Results

Two-player Leduc Poker

In Figure 2, we test our algorithm on two-player Leduc poker and plot the regret curves (w.r.t the full game) given by FP, DO, PSRO with PRD, and PSRO with RRD (under two stopping criteria). We first observe that RRD yields a rapid convergence to a low-regret value compared to other MSSs. It is quite striking that RRD outperforms PRD (prior best known for this game) by such a large margin.

To show the benefits of using a regret threshold as a stopping criterion compared to a fixed number of RD updates, we plot the best regret curve of RRD using a fixed number of RD updates. We observe that RRD performs better using a regret threshold. This is because the number of RD updates that produces the right level of regularization varies across empirical games.

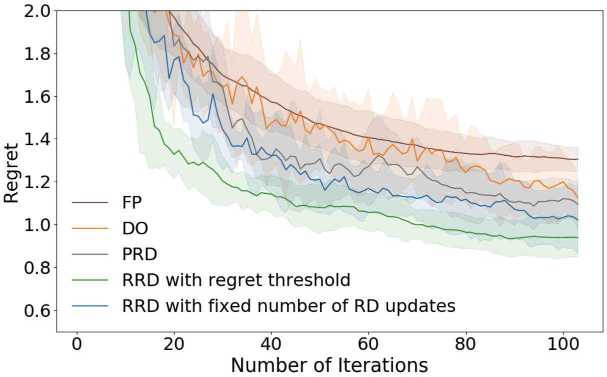


Figure 2 RRD performance in two-player Leduc Poker.

Real-World Games

We further evaluate our algorithms in four of the "real-world games" studied by Czarnecki *et al.* [2020]: Hex, Connect four, Misere Tic Tac Toe, and Go. We observe that RRD exhibits faster convergence than FP and DO in all four games. We report further experimental details and full game descriptions in Appendix B.2.

Multi-Player Games

We apply the combination of BPS and RRD to three-player Leduc poker. As shown in Figure 3, although RRD is applied only to the subgame containing the support of exact equilibrium, learning still benefits from regularization. In Table 1, we list the average of number of profiles evaluated in the empirical game at different PSRO iterations with and without BPS. To evaluate a profile, we estimate payoffs by averaging over 1000 samples. Employing BPS in this example saved approximately 10% simulation effort, compared to exhaustive estimation. We can also see that evaluation savings become somewhat more significant as the number of iteration increases. For example, from iterations 41 to 50 we evaluated 53890 profiles, while the profile space increased by 61000, a savings of 11.7%. Experience in prior EGTA work demonstrates that the fractional savings are substantially greater for games with more than three players [Brinkman and Wellman, 2016].

Attack-Graph Games

Attack graphs [Miehling *et al.*, 2015] are a tool in cybersecurity analysis employed to model the paths by which an adversary may compromise a system. An *attack-graph game* is a two-player general-sum game defined on the attack graph where an attacker attempts to compromise a sequence of nodes to reach *goal* nodes and a defender endeavors to protect any node (e.g., deny an access). Reaching the *goal* nodes within a finite horizon provides a large benefit for the attacker and a substantial loss for the defender. Both offensive and defensive actions are associated with a cost. The ability of the attacker to choose any subset of feasible nodes and of the defender to defend any subset of the nodes induces action spaces of combinatorial size.

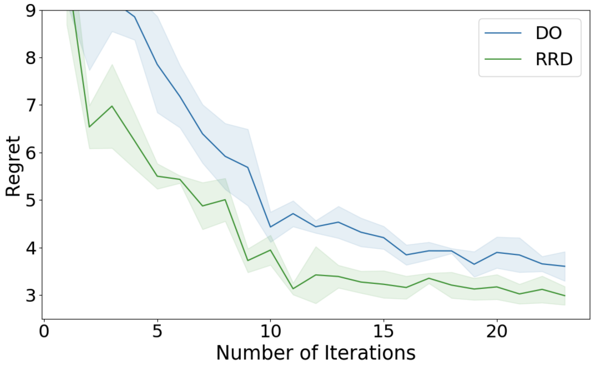


Figure 3 RRD performance in three-player Leduc poker.

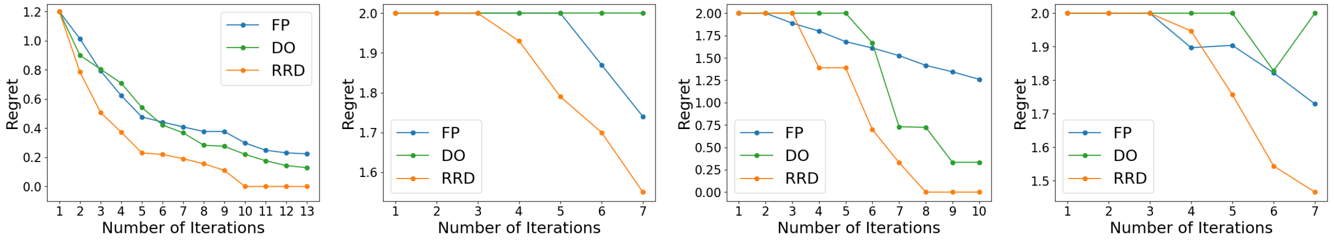
Iter#	#Profiles w. BPS	#Profiles	Savings Pct.
10	880	1000	12.0%
20	7100	8000	11.1%
30	24667	27000	7.5%
40	58400	64000	8.8%
50	112290	125000	11.7%

Table 1 Savings in profile simulation due to BPS in a three-player game.

In Figure 5, we show the performance of RRD on an attack-graph game instance with 100 nodes and hence 2^{100} possible combinatorial actions. Since the game is too large to analyze exhaustively, we first construct a particular set of deep Q-network (DQN) [Mnih *et al.*, 2013] strategies with 125 strategically-diverse strategies in total, following the strategy sampling approach by Czarnecki *et al.* [2020]. Then we apply game-theoretic analysis (i.e., FP, DO, and RRD) to this set of strategies. Each regret curve is an average over 5 randomly-selected initial strategies. From Figure 5, we observe that even though the game of interest is large and beyond two-player zero-sum games, RRD still boosts faster convergence and less variance than DO and FP.

Bargaining Games

We consider a non-zero-sum incomplete-information bargaining game, in which two players engage in a sequential process to reach a deal over a vector of items [Lewis *et al.*, 2017]. In our setting, there are three items, and up to $T = 10$ rounds of offers. The value of each item is sampled from a commonly known distribution for both players and each player only knows their own value realizations. If a deal is



(a) Hex. (b) Connect four. (c) Misere Tic Tac Toe. (d) Go (size=4).

Figure 4 RRD performance compared to FP and DO in four games studied by Czarnecki *et al.* [2020].

reached at time $t \leq T$, players receive their corresponding values for the items specified, discounted by a factor of γ^t ($\gamma = 0.9$). If no deal is reached, both players receive zero utility. We represent negotiation policies using a neural network, and employ DQN to compute approximate best responses.

In Figure 6, we consider social welfare (SW) as a performance measure and compare the averaged social welfare of the same solution concept in the empirical games given by different MSSs (i.e., RRD, DO, FP, MWCCE, and MGCCE). Each SW value is an average over 5 runs and we show the standard deviations in the Appendix B.3. We select five solution concepts for performance comparison, including maximum SW pure strategy profile (Max SW), NE, uniform distribution over strategies, MWCCE, and MGCCE. From Figure 6, we observe that RRD can generate all five solutions with higher social welfare than others. This observation also indicates that the best MSS to approximate a solution (e.g., NE, MWCCE, and MGCCE) may not be the one that uses the solution concept directly as a best response target.

5.2 Stability with Varying Regret Threshold

To investigate the stability of learning performance w.r.t the regret threshold λ, we select a wide range of λs for RRD and compare the regrets at the last iteration of PSRO under these λs with the regret of DO in two-player Leduc poker. We plot the regrets in Figure 7a. From Figure 7a, we observe that all λs in the range yield a better learning performance than DO, which demonstrates the stability of the performance of RRD w.r.t the regret threshold λ. In addition, we observe that as the value of regret threshold λ increases, the learning performance first improves and then becomes worse. This means that either excessive or inadequate regularization would damage the overall learning performance.

6 Explaining the Performance of MSSs

Prior studies of strategy exploration recognized that best-responding to exact NE is not ideal, and demonstrated improvements over DO through alternative MSSs or other approaches. However, to date there has been no satisfactory explanation for what makes for an effective best-response target. We briefly discuss some prior studies, then provide a novel insight—based on our experimental observations—which helps to explain why regularization is helpful in this context.

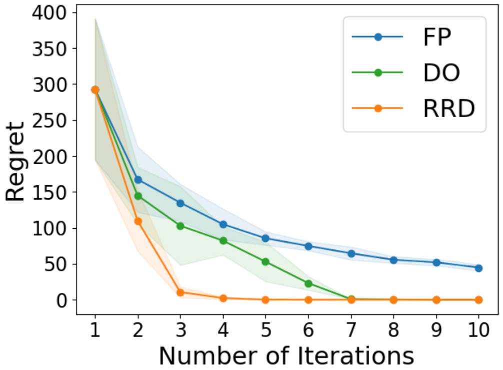


Figure 5 RRD outperforms FP and DO in the attack-graph game.

6.1 Prior Regularization-Related Approaches

In an early study of strategy exploration, Schvartzman and Wellman [2009a] found that adding noise to NE alters the path of equilibrium search and accelerated the overall learning. But how this noise contributed to the acceleration was unexplained. As noted above, the originators of PSRO introduced PRD expressly to address overfitting to NE [Lanctot *et al.*, 2017]. PRD promotes exploration by ensuring each strategy in the empirical game a fixed minimal probability in the best-response target. With similar motivation, Wang *et al.* [2019] proposed to alternate FP and DO randomly, and Wright *et al.* [2019] adjusted the best response to NE by tuning against previous opponents.

Related to this line of work, Balduzzi *et al.* [2019] introduced the term ***Gamescape*** to describe the conceptual strategy space covered by the empirical game. They proposed an MSS called ***rectified Nash***, designed to qualitatively extend the Gamescape. However, the concept of Gamescape was described primarily through illustration in a particular simple game, which is instructive but does not provide operational definitions or measures.

6.2 A Novel Explanation

Our key insight is that the performance of strategy exploration is strongly related to the regret of best-response targets *w.r.t the full game*. To illustrate this phenomenon, Figure 7b presents regret curves for PSRO with RRD in two-player Leduc poker. The two curves show true-game regret

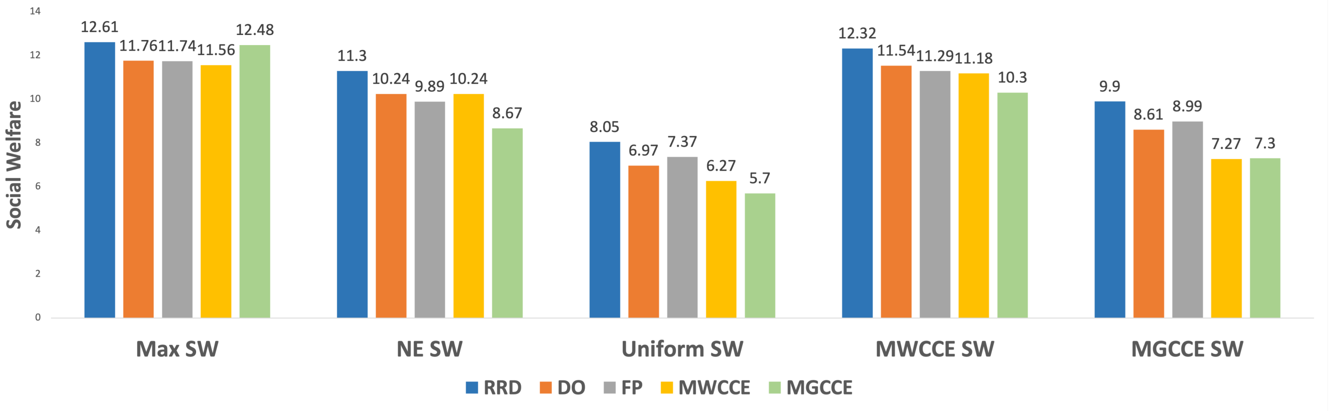
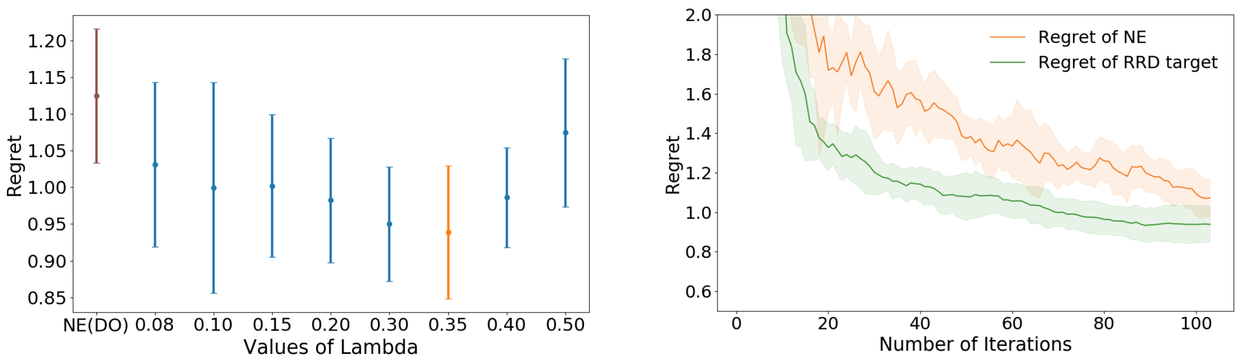


Figure 6 RRD performance in bargaining games. Each color represents an MSS and each bundle of colors shows the SW of a given solution concept in the corresponding empirical games.



(a) Range of regret thresholds.

(b) Decreased regret when regularization is applied.

Figure 7 Properties of learning with RRD in two-player Leduc Poker.

for NE and RRD, respectively, as computed at each PSRO iteration. Note that throughout the run, the regret of the RRD solution is much smaller than that of the empirical NE. We observe the same phenomenon in PSRO runs generated by other MSSs (see Appendix B.4). In other words, whereas RRD has higher regret than NE in the empirical game (λ versus zero), it reliably has lower regret in the full game. Since our ultimate objective is a full-game low-regret solution, this helps to explain why the regularization imposed by RRD apparently provides robustly improved performance for strategy exploration.

Note that this observation only goes so far; it is not the case that minimizing full-game regret always provides the optimal best-response target for strategy exploration. This is because the lowest full-game regret profile may not change much from one PSRO iteration to the next, and so selecting targets on that basis may compromise the diversity of constructed empirical games. We tested this explicitly by using as an MSS the profile in the empirical game that has the lowest regret with respect to the full game. Our results confirm that the extreme choice of target is indeed suboptimal for strategy exploration (details provided in Appendix C.2).

7 Conclusion

We propose RRD as a novel MSS for PSRO, explicitly based on regularization. By controlling the regret threshold, the degree of regularization can be adjusted to suit a particular strategy exploration context. In our experiments, we show that RRD outperforms several existing MSSs in various games and investigate many properties of learning with RRD. To help scale beyond two-player games, we propose BPS, a PSRO-compatible profile search method that avoids exhaustive simulation of the game matrix. We show the benefit of regularization when combining BPS with RRD in three-player Leduc poker. Finally, we demonstrate that the performance of strategy exploration is strongly related to the regret of best-response targets and regularization could significantly decrease the regret of best-response targets, thus contributing to an improved learning.

References

[Balduzzi *et al.*, 2019] David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech M Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. In *36th International Conference on Machine Learning*, pages 434–443, 2019.

[Brinkman and Wellman, 2016] Erik Brinkman and Michael P. Wellman. Shading and efficiency in limit-order markets. In *IJCAI-16 Workshop on Algorithmic Game Theory*, 2016.

[Brown, 1951] George W. Brown. Iterative solution of games by fictitious play. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 374–376. Wiley, 1951.

[Czarnecki *et al.*, 2020] Wojciech Marian Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. Real world games look like spinning tops. In *34th Conference on Neural Information Processing Systems*, 2020.

[Dinh *et al.*, 2022] Le Cong Dinh, Stephen Marcus McAleer, Zheng Tian, Nicolas Perez-Nieves, Oliver Slumbers, David Henry Mguni, Jun Wang, Haitham Bou Ammar, and Yaodong and Yang. Online double oracle. *Transactions on Machine Learning Research*, October 2022.

[Fearnley *et al.*, 2015] John Fearnley, Martin Gairing, Paul Goldberg, and Rahul Savani. Learning equilibria of games via payoff queries. *Journal of Machine Learning Research*, 16:1305–1344, 2015.

[Jordan *et al.*, 2008] Patrick R. Jordan, Yevgeniy Vorobeychik, and Michael P. Wellman. Searching for approximate equilibria in empirical games. In *7th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 1063–1070, 2008.

[Jordan *et al.*, 2010] Patrick R. Jordan, L. Julian Schvartzman, and Michael P. Wellman. Strategy exploration in empirical games. In *9th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 1131–1138, 2010.

[Lanctot *et al.*, 2017] Marc Lanctot, Vinicius Zambaldi, Audrūnas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *31st Annual Conference on Neural Information Processing Systems*, pages 4190–4203, 2017.

[Lanctot *et al.*, 2019] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, et al. OpenSpiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*, 2019.

[Lewis *et al.*, 2017] Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? End-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*, 2017.

[Marris *et al.*, 2021] Luke Marris, Paul Muller, Marc Lanctot, Karl Tuyls, and Thore Graepel. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. In *38th International Conference on Machine Learning*, pages 7480–7491, 2021.

[McAleer *et al.*, 2021] Stephen McAleer, John Lanier, Pierre Baldi, and Roy Fox. CFR-DO: A double oracle algorithm for extensive-form games. In *AAAI-21 Workshop on Reinforcement Learning in Games*, 2021.

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

[McKelvey and Palfrey, 1998] Richard D. McKelvey and Thomas R. Palfrey. Quantal response equilibria for extensive form games. *Experimental Economics*, 1(1):9–41, 1998.

[McKelvey *et al.*, 2006] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. Gambit: Software tools for game theory, 2006.

[McMahan *et al.*, 2003] H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *20th International Conference on Machine Learning*, pages 536–543, 2003.

[Miehling *et al.*, 2015] Erik Miehling, Mohammad Rasouli, and Demosthenis Teneketzis. Optimal defense policies for partially observable spreading processes on Bayesian attack graphs. In *2nd ACM Workshop on Moving Target Defense*, pages 67–76, 2015.

[Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[Muller *et al.*, 2020] Paul Muller, Shayegan Omidshafiei, Mark Rowland, Karl Tuyls, Julien Perolat, Siqi Liu, Daniel Hennes, Luke Marris, Marc Lanctot, Edward Hughes, et al. A generalized training approach for multiagent learning. In *8th International Conference on Learning Representations*, 2020.

[Omidshafiei *et al.*, 2019] Shayegan Omidshafiei, Christos Papadimitriou, Georgios Piliouras, Karl Tuyls, Mark Rowland, Jean-Baptiste Lespiau, Wojciech M. Czarnecki, Marc Lanctot, Julien Perolat, and Remi Munos. α-rank: Multi-agent evaluation by evolution. *Scientific Reports*, 9(1):1–29, 2019.

[Phelps *et al.*, 2006] S. Phelps, M. Marcinkiewicz, S. Parsons, and P. McBurney. A novel method for automatic strategy acquisition in n-player non-zero-sum games. In *5th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 705–712, 2006.

[Schvartzman and Wellman, 2009a] L. Julian Schvartzman and Michael P. Wellman. Exploring large strategy spaces in empirical game modeling. In *AAMAS-09 Workshop on Agent-Mediated Electronic Commerce*, 2009.

[Schvartzman and Wellman, 2009b] L. Julian Schvartzman and Michael P. Wellman. Stronger CDA strategies through empirical game-theoretic analysis and reinforcement learning. In *8th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 249–256, 2009.

[Smith and Price, 1973] J. Maynard Smith and George R. Price. The logic of animal conflict. *Nature*, 246(5427):15–18, 1973.

[Sureka and Wurman, 2005] Ashish Sureka and Peter R. Wurman. Using tabu best-response search to find pure strategy Nash equilibria in normal form games. In *4th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1023–1029, 2005.

[Taylor and Jonker, 1978] Peter D. Taylor and Leo B. Jonker. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40(1-2):145–156, 1978.

[Tuyls *et al.*, 2020] Karl Tuyls, Julien Pérolat, Marc Lanctot, Edward Hughes, Richard Everett, Joel Z. Leibo, Csaba Szepesvári, and Thore Graepel. Bounds and dynamics for empirical game theoretic analysis. *Autonomous Agents and Multi-Agent Systems*, 34:7, 2020.

[Wang *et al.*, 2019] Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, and Fei Fang. Deep reinforcement learning for green security games with real-time information. In *33rd AAAI Conference on Artificial Intelligence*, pages 1401–1408, 2019.

[Wang *et al.*, 2022] Yongzhao Wang, Qiurui Ma, and Michael P. Wellman. Evaluating strategy exploration in empirical game-theoretic analysis. In *23th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 1346–1354, 2022.

[Wellman, 2016] Michael P. Wellman. Putting the agent in agent-based modeling. *Autonomous Agents and Multi-Agent Systems*, 30:1175–1189, 2016.

[Wright *et al.*, 2019] Mason Wright, Yongzhao Wang, and Michael P. Wellman. Iterated deep reinforcement learning in games: History-aware training for improved stability. In *20th ACM Conference on Economics and Computation*, pages 617–636, 2019.

A Table of Acronyms

Abbreviation	Definition
BPS	Backward Profile Search
DO	Double Oracle
EGTA	Empirical Game-Theoretic Analysis
FP	Fictitious Play
MSS	Meta-Strategy Solver
MRCP	Minimum Regret Constrained Profile
NE	Nash Equilibrium
PRD	Projected Replicator Dynamics
PSRO	Policy Space Response Oracle
QRE	Quantal Response Equilibrium
RD	Replicator Dynamics
RL	Reinforcement Learning
RRD	Regularized Replicator Dynamics

Table 2 Table of acronyms in alphabetical order.

B Extra Experimental Results

B.1 Advantage over RD with a Fixed Number of iterations

In Figure 8, we show the number of RD updates needed to reach the regret threshold λ. We fix the lower bound of the number of iterations to 10000. As shown in Figure 8, the number of RD updates required to reach the regret threshold λ varies across PSRO iterations, which again emphasizes the need of adjusting the number of RD updates dynamically rather than fixing the number of RD updates. Moreover, it is interesting to observe that the number of iterations is higher in the middle of the learning while it is lower at both the beginning and the end. This contradicts the stereotype that RD needs more iterations to converge in an empirical games with more diverse strategies.

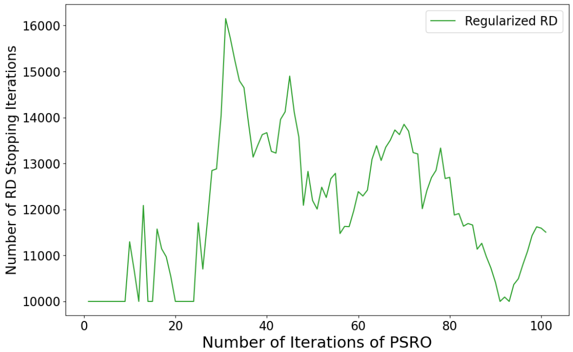


Figure 8 The number of RD updates reaching the regret threshold λ at different PSRO iterations.

B.2 Game Descriptions

Real-world Game Instances

The real-world games in our experiments are game models distilled from the full games by Czarnecki *et al.* [2020]. The descriptions of the games are shown as below.

- Hex is a two-player board game, in which players attempt to connect opposite sides of a hexagonal board.
- Connect four is a two-player connection board game, in which the players choose a color and then take turns dropping colored tokens into the board. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own tokens.
- Misere Tic Tac Toe is a variant of Tic Tac Toe where one wins if and only if the opponent makes a line rather than itself.
- Go with size 4 is a Go game with a smaller board than the original Go.

Experimental Details for Real-World Games

We evaluate all MSSs following the consistency metric by Wang *et al.* [2022], which measures the quality of intermediate game models. In all experiments, each player is initialized a set of poor-performing strategies at the beginning. The set of initial strategies is identical for all MSSs. Since these games are represented as matrix games, both best response computation and regret computation are deterministic. Hence no error bar is reported. For RRD, we set a regret threshold $\lambda = 0.5$ for all experiments.

Experimental Details for Bargaining Games

We follow the bargaining game model by Lewis *et al.* [2017]. Each player is given a uniformly generated value function, which gives a non-negative value for each item. The value functions are constrained so that: (1) the total value for a user of all items is 10; (2) each item has non-zero value to at least one user; and (3) some items have non-zero value to both players. These constraints enforce that it is not possible for both players to receive a maximum score, and that no item is worthless to both players, so the negotiation will be competitive. After 10 turns, players are able to complete the negotiation with no agreement, which is worth 0 points to both players. For RRD, we set a regret threshold $\lambda = 0.3$ for all experiments.

Experimental Details for Attack-graph Games

We follow the attack graph model of Miehling *et al.* [2015]. As described in the main paper, an attack graph is given by a DAG $\mathcal{G} = (V, E)$, where vertices $v \in V$ represent security conditions and edges $e \in E$ are labelled by exploits. An *attack-graph game* is defined by an attack graph endowed with additional specifications. The game is a two-player partially observable stochastic game that is played over a finite number of time steps T. At each time step t, the state of the game is given by the state of the graph, which is simply whether nodes are active or not (i.e., compromised by attacker or not), indicated by $s_t(v) \in \{0, 1\}$. It is assumed to be fully observable for the attacker while the defender receives a noisy observation $o_t(v) \in \{0, 1\}$ of the state, based on commonly known probabilities $P_v(o = 0 \mid s = 1)$ and $P_v(o = 1 \mid s = 0)$ for each node v. Positive observations are called *alerts*.

The attacker and defender act simultaneously at each time step t. The defender's atomic action is to defend a node, thus, the atomic actions are simply V. The defender's action space for any time step is 2^V, meaning it can choose to defend any subset of the nodes. The attacker's atomic action set varies with time and is based on current graph state. Exploits on an edge are feasible only if the origin node of the edge is activated. Nodes without parents, called root nodes, can be attacked without preconditions. For the special case of attacking an *AND* node, the attacker's atomic action is treated as attacking a node rather than choosing exploits on all incoming edges. Thus, the attacker's atomic actions can be viewed as selecting edges (feasible exploits) for attacking *OR* nodes or selecting nodes from *AND* nodes whose parent nodes are all active. The attacker's action space at any time step is the power set of feasible atomic actions.

Defender actions override attacker actions, that is, any node v that is defended becomes inactive. Otherwise, active nodes remain active; an *AND* node v that is attacked becomes active with probability $P(v)$, and any *OR* node becomes active based on the success probabilities $P(e)$ of attacked edges.

Each *goal* node, v, carries reward $R_A(v)$ for attacker and penalty $P_D(v)$ for defender for all time steps in which v is active. Any atomic action a of an agent has a cost: $c_{a,D}(v)$ for nodes defended in case of defender; $c_{a,A}(v)$ for *AND* nodes selection and $c_{a,A}(e)$ for edges selection in case of attacker. For simplicity, we omit the argument (v or e) for action a in the notation. When obvious from context we also drop the subscript D and A, simply using v_a and c_a to denote the target node of a and the cost of action a respectively.

The defender's loss (negative reward) at any time step is the cost of all its atomic actions (i.e., total cost of nodes defended), plus the penalty for goal nodes active after the moves. The defender's long-term payoff is the discounted expected sum of losses over time. Similarly, the attacker's long-term payoff is the discounted expected sum of payoff per time step, where the per time step payoff is the reward for active goal nodes minus the cost of atomic actions used in that time step.

A policy for either player (pure strategy in the game) maps its observations at any step to a set of actions. For the attacker, the mapping is from states to action sets. The defender only partially observes state, so its policy maps observation histories to action sets. In our implementation, we limit the defender to a fixed length h of past observations for tractability. Solving the game means finding a pair of mixed strategies (distribution over pure strategies), one for each player, that constitutes a NE.

B.3 Standard Deviations of Social Welfare in Bargaining Games

In Table 3, we list the standard deviation of social welfare for difference solutions. This corresponds to Figure 6 in the main paper.

MSSs	Max SW	NE	Uniform	MWCCE	MGCCE
RRD	0.60	1.14	1.40	0.73	1.05
DO	0.73	1.11	1.39	0.77	0.98
FP	0.80	1.58	1.63	1.01	1.53
MWCCE	0.67	1.11	1.66	0.70	2.49
MGCCE	1.00	1.78	1.60	1.11	1.88

Table 3 Table of standard deviations for bargaining games.

B.4 Decreased Regret by Regularization in PSRO with NE

In Figure 9, we run PSRO with NE in two-player Leduc poker and plot the regret of the best-response targets (w.r.t the true game) given by RRD and the regret of NE (w.r.t the true game) in the intermediate empirical games. From Figure 9, we observe that as in PSRO with RRD (Figure 7b), given an intermediate empirical game, the regret of best-response targets again decreases after applying regularization.

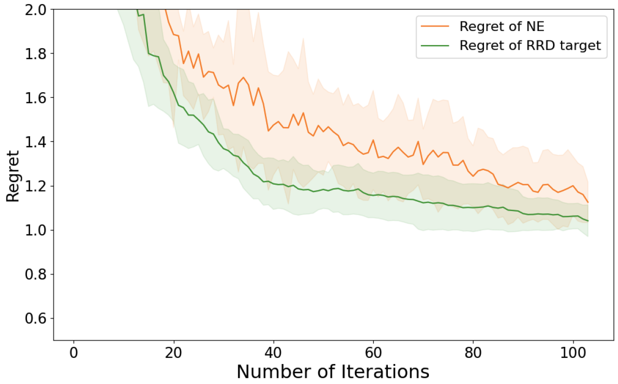


Figure 9 Decreased regret when regularization is applied in PSRO run with NE as MSS.

B.5 Performance Check with Consistency Evaluation Solver

In Figure 10, we follow the rule of consistency [Wang *et al.*, 2022], a critical measure for the quality of generated empirical games. According to the consistency criterion, we compare MSSs with the same RRD-based regret and authenticate the faster improvement of RRD over DO and PRD in terms of the quality of generated empirical games.

B.6 Experimental Parameters

We use OpenSpiel Lanctot *et al.* [2019] default parameter sets for experiments on Leduc: each payoff entry in an empirical game is an average of 1000 repeated simulations; DQN is adopted as a best response oracle, its parameters are shown in Table 4. The poker games are asymmetric in the sense that one player always moves first.

PRD is implemented with lower bound for strategy probability 1e-10, maximum number of steps 1e5 and step size 1e-3. RD shares the same step size but a varying number of

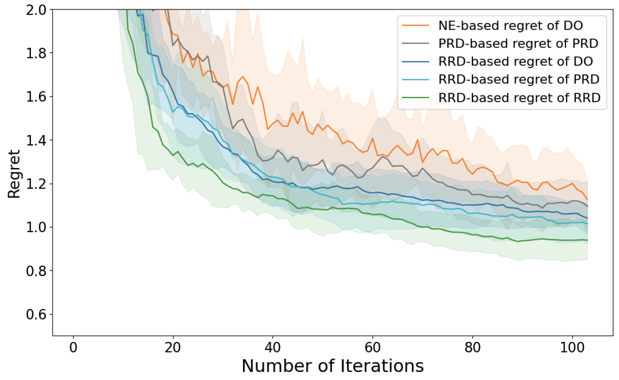


Figure 10 RRD performance in two-layer Leduc poker with a consistency measure by Wang *et al.* [2022].

Parameter	Value
learning rate	1e-2
Batch Size	32
Replay Buffer Size	1e4
Episodes	1e4
optimizer	adam
layer size	256
number of layer	4
Epsilon Start	1
Epsilon End	0.1
Exploration Decay Duration	3e6
discount factor	0.999
network update step	10
target network update steps	500

Table 4 DQN hyperparameters

steps controlled by the regret threshold λ. We test the learning performance of RRD with λ ranging from 0 to 0.6. We get best learning performance with $\lambda = 0.35$ in Leduc poker. In three-player Leduc poker, we experiment with $\lambda = 0.6$.

#Nodes	100
Costs	Uniform in [0, 1] for attacker; Uniform in [2, 4] for defender
Rewards	Uniform in [10, 20]
Penalties	Uniform in [7, 10]
#Goal Nodes	6
Activation Prob.	Uniform in [0.6, 1]
False Alarm Prob.	Uniform in [0, 0.2]

Table 5 Attack-graph game hyperparameters.

B.7 Code Availability

Following the double-blind policy, we will publish our code after the review.

B.8 Computational Resources

We used one 2 × 3.0 GHz Intel Xeon Gold 6154 CPU with 16gb memories.

C PSRO with Other Novel MSSs

C.1 Results for QRE being an MSS

One common assumption in game-theoretic analysis is the rationality of players (i.e., players act according to NE). Since our regularization approach prevents players from playing NE to some extent within the empirical game, it can be viewed as a way of restricting the rationality of players, which naturally relates our approach to Quantal Response Equilibrium (QRE) [McKelvey and Palfrey, 1995; McKelvey and Palfrey, 1998], an equilibrium notion with bounded rationality. One common specification for QRE is logit equilibrium in which players' strategies take the following form

$$\sigma_i(s_i) = \frac{exp(\tau u_i(s_i, \sigma_{-i}))}{\sum_{s_i' \in S_i} exp(\tau u_i(s_i', \sigma_{-i}))}, \forall s_i \in S_i, i \in N.$$

where τ is a parameter governing the rationality of players.

To see the performance of QRE being an MSS, we evaluate the learning performance of PSRO with QRE. Specifically, we compute the QRE of the empirical game at every PSRO iteration using Gambit [McKelvey *et al.*, 2006] and analyze the learning performance of QRE.

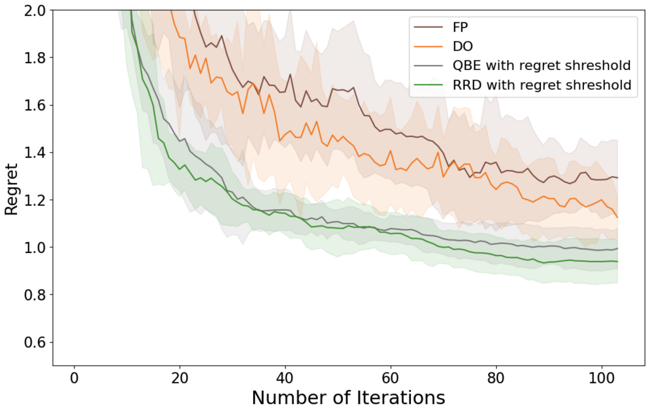


Figure 11 Performance of PRSO with QRE in two-layer Leduc poker.

Figure 11 shows the learning performance in Leduc poker with QRE as an MSS. For comparison, we also plot the learning curve of RRD with the same regret threshold of QRE. Although QRE shows a slight divergence in the end, it still demonstrates the potential of using QRE as an MSS in PSRO.

C.2 Results for Using MRCP as an MSS

Definition of MRCP

The profile in the empirical game closest to being a solution of the full game is called ***minimum-regret constrained-profile*** (MRCP) [Jordan *et al.*, 2010]. Formally, $\bar{\sigma}$ is an MRCP iff:

$$\bar{\sigma} = \operatorname*{argmin}_{\sigma \in \Delta(X)} \sum_{i \in N} \rho_i^{\mathcal{G}}(\sigma)$$

The regret of MRCP thus provides a natural measure of how well X covers the strategically relevant space [Wang *et al.*, 2022].

Experiments of Learning with MRCP

We have observed the existence of strategy profiles with lower global regret than NE in the empirical game and the experimental results of regularization shows that training against them results in improved learning performance than DO. One natural question to ask is whether training against the most stable profile targets can benefit strategy exploration the most (e.g., using MRCP as MSS).

To answer this question, we compare the performance of MRCP as MSS against DO and FP in the matrix-form two-player Kuhn's poker and a synthetic two-player zero-sum game. In Kuhn's poker, we randomly select 4 starting points and implement PSRO. Fig. 12a-12d show that with 3 out of 4 starting points, MRCP converges slight faster than DO. For the matrix game, Fig. 12e and 12f show the benefits of applying MRCP but the performance varies across different starting points.

In Fig. 12, we observe that the MRCP has some power for heuristic strategy generation. However, the advantage of using MRCP is not satisfactory in terms of convergence rate and computational complexity. We also find that using MRCP may converge slower in other games like Blotto compared to DO and PRD.

The experiments show that training against the lowest-regret profile in the empirical game does not necessarily lead to a better overall learning performance. This is because the lowest-regret profile in the empirical game may not be changed a lot after adding a new strategy to the empirical game, which yields similar strategies continued to be added over PSRO iterations. Continuing adding similar strategies would result in only little performance improvement over PSRO iterations.

Now we illustrate why pursuing best-response targets with extremely low regret may result in slow learning using a matrix game shown in Table 7. The matrix game contains 1000 strategies for each player. All missing entries of the payoff matrix are $(0, 0)$. Let's start PSRO with the first strategy (s_1, s_1). This matrix game is designed to have long equilibrium search path for DO (as in many real-world games). Specifically, by best-responding to (s_1, s_1), each player adds $s2$ to the empirical game whose new NE is (s_2, s_2). Similarly, if we best-respond to the equilibrium at each PSRO iteration, we would first get a new NE (s_3, s_3) and then a long equilibrium path through the diagonal until we reach the NE of the full game (s_{1000}, s_{1000}).

Without loss of generality, suppose we are at iteration 2 (i.e., the empirical game includes (s_1, s_2) and (s_2, s_2) is an empirical NE). The MRCP of this empirical (symmetric) game is approximately $(1s1, 0s2)$ with regret 0.0112=0.022 (sum over players) (the regret of accurate MRCP is even lower). The regret of empirical NE is $0.1 * 2 = 0.2$ by deviating to s_3 from (s_2, s_2). When best responding to MRCP, we add s_{500} (only considering deviation strategies outside the empirical) and then MRCP remains the same (i.e., $(1s_1, 0s_2)$) for the empirical game. Therefore, further best responding to the MRCP may again add some strategies similar to s_{500} and may not improve the learning performance dramatically.

Suppose RRD gives probability $(0.5, 0.5)$ on (s_1, s_2), then best responding to $(0.5s_1, 0.5s_2)$ leads to equilibrium strategy s_{1000} directly, jumping out of the long equilibrium path of DO. The regret of $(0.5s_1, 0.5s_2)$ is $(0.005 \times 0.5 + 0.199 \times 0.5 - 0.011 \times 0.25 - 0.1 * 0.25) \times 2 = (0.102 - 0.02775) \times 2 = 0.074252 = 0.1485$ (0.02 (regret of MRCP) $<$ 0.1485 (regret of RRD) $<$ 0.2 (regret of NE)).

In this example, best responding to the relatively low full-game regret profile, RRD avoids falling into the long diagonal path as DO. Meanwhile, its regret is not as low as the regret of MRCP so that the best-response target at each PSRO iteration would keep being updated significantly rather than staying similarly.

Extra Properties of Learning with MRCP

Theoretically, multiple MRCPs could exist in an empirical game and MRCP is not necessarily a pure strategy profile in general. Moreover, purely using MRCP as an MSS does not guarantee convergence to NE since the best-responding strategy could already exist in the empirical game. We define this property of MRCP as follows.

Definition. An empirical game with strategy space $X \subseteq S$ is ***closed*** with respect to MRCP $\bar{\sigma}$ if

$$\forall i \in N, s_i = \operatorname*{argmax}_{s_i' \in S_i} u_i(s_i', \sigma_{-i}) \in X_i.$$

To illustrate this concept, consider the symmetric zero-sum matrix game in Table 6. Starting from the first strategy of each player and implementing PSRO with MRCP, we have the empirical game including a^1 and a^2. Since the (a_1^1, a_2^1) is a MRCP (considered all pure and mixed strategy profiles) and best responding to the profile gives a^2 again, the empirical game is *closed* and never extends to the true game wherein the true NE is (a_1^3, a_2^3).

	a_2^1	a_2^2	a_2^3
a_1^1	(0, 0) [2]	(-1, 1) [6]	(-0.5, 0.5)
a_1^2	(1, -1) [6]	(0, 0) [10]	(-5, 5)
a_1^3	(0.5, -0.5)	(5, -5)	(0, 0)

Table 6 Symmetric Zero-Sum Game for MRCP. Regret of profiles is shown in the square parenthesis.

In our experiments, we deal with this issue by only introducing new strategy with highest deviation payoff outside the empirical game and thus guarantee convergence. An alternative is to switch between DO and MRCP whenever this issue happens and the convergence is guaranteed due to the convergence property of DO.

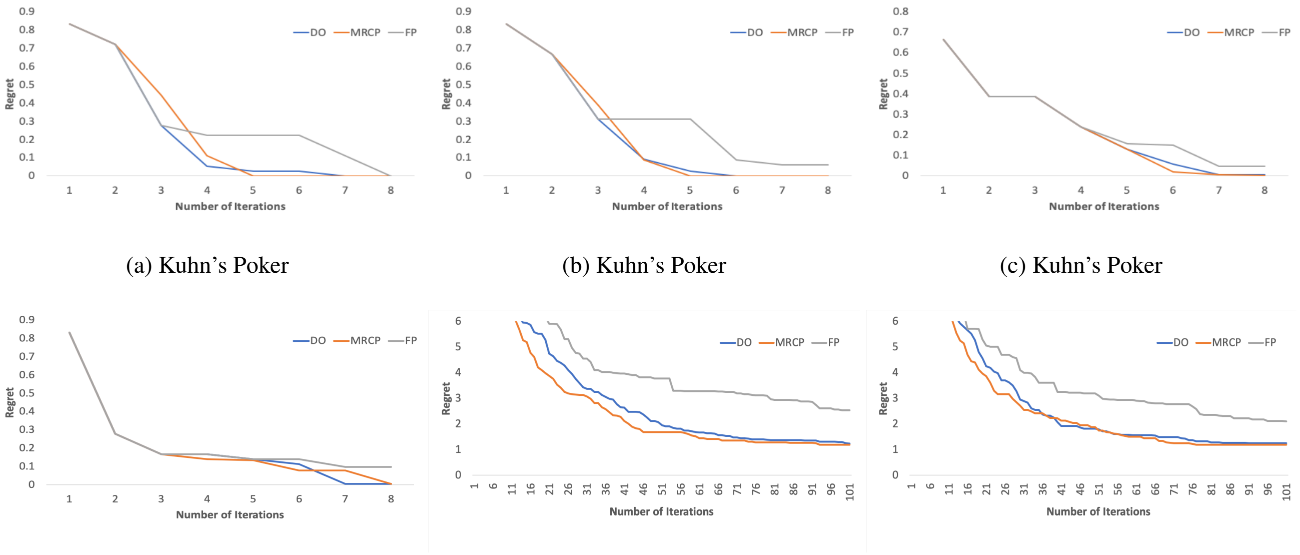
D Convergence of RRD

D.1 Proof of Convergence

We first define the concept ϵ-***closeness***, which can be viewed as a stopping condition of PSRO.

Definition (ϵ-closeness)**.** An empirical game with strategy space $X \subseteq S$ is ϵ-***closed*** with respect to certain ϵ-NE $\sigma \in \Delta(X)$ and operator o if and only if $o(\sigma) \in X$.

For example, if o is a best-response operator and $\epsilon = 0$, this definition means there is no beneficial deviation from the



(a) Kuhn's Poker

(b) Kuhn's Poker

(c) Kuhn's Poker

(d) Kuhn's Poker

(e) Zero-Sum Game

(f) Zero-Sum Game

Figure 12 Performance of using MRCP as an MSS. Y axis depicts MRCP-based regret.

NE σ of the empirical game, and thus σ is a NE of the full game. When $\epsilon \neq 0$, ϵ-closeness indicates that the deviation strategy of the ϵ-NE σ of the empirical game already exists in the empirical game. Note that there could exist an infinite number of ϵ-NE in an empirical game given a specific ϵ, so the definition of ϵ-closeness is associated with a specific ϵ-NE.

Next we prove that if an empirical game is ϵ-closed with respect to certain ϵ-NE $\sigma \in \Delta(X)$ and best-response operator o, then σ is an ϵ-NE of the full game.

Lemma 1. *If an empirical game with strategy space $X \subseteq S$ is ϵ-closed with respect to certain ϵ-NE $\sigma \in \Delta(X)$ and best-response operator o, then σ is an ϵ-NE of the full game $\mathcal{G}$.*

Proof. Since σ is an ϵ-NE in the empirical game, there is no deviation strategy within the empirical game that results in regret large than ϵ. Mathematically, we have $\forall i \in N$, $\max_{s'_i \in X_i} u_i(s'_i, \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \leq \epsilon$. Since the best-response operator finds the best deviation w.r.t the true game and the best deviation falls into the empirical game, we have $\forall i \in N$, $\max_{s'_i \in S_i} u_i(s'_i, \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \leq \epsilon$. Then σ is an ϵ-NE of the full game $\mathcal{G}$. □

Consider the finite strategy space S. We prove the following theorem that if we train against an ϵ-NE of the empirical game at each iteration of PSRO, we end up with an empirical game containing at least one ϵ-NE. By setting ϵ to be a reachable λ, we prove the Theorem 1.

Theorem 1. *Assuming the access to an exact best response oracle, Policy Space Response Oracle with Regularized Replicator Dynamics associated with a reachable regret threshold λ converges to an empirical game containing at least one λ-NE.*

Proof. Since we have finite strategy space S, ϵ-closeness with respect to certain σ is always reachable by training against an ϵ-NE at each iteration. Once the ϵ-closeness is reached, the corresponding σ is an ϵ-NE of the full game due to Lemma 1. Due to the population property of PSRO [Wang *et al.*, 2022], profiles with lower regret than ϵ could also exist. □

This result generalizes DO and its convergence guarantee to scenarios where training target is not strictly restricted to NE. Note that similar results have been proved in earlier works [Dinh *et al.*, 2022; McAleer *et al.*, 2021]. However, they miss the fundamental fact of an empirical game, that is, an empirical game creates a profile space, within which profiles with regret much smaller than the target profile could exist. This fact reveals one major advantage of using game models to facilitate game learning since with a game model, we simply need to capture a NE within the strategy space rather than requiring a sequence of profiles converge to NE as in the online learning setting.

D.2 Performance Confusion

A common confusion is why RRD is useful in the sense that RRD has weaker theoretical convergence guarantee than DO. Note that the proof for the convergence of DO by McMahan *et al.* [2003] describes whether it converges rather than how fast it converges. In the worst case, DO needs to add all strategies (assuming the game is finite) to the empirical game to reach the convergence, which is apparently trivial. In games of interest (e.g., Leduc poker), the strategy space is usually very large and cannot be enumerated. Despite algorithms like DO guarantee a convergence to NE, it is possible that such algorithms need to explore a large number of strategies to reach the convergence, which is far beyond the computational budget. The goal of strategy exploration is to learn towards a NE as close as possible within the budget. Therefore, strategy exploration focuses more on the learning performance of

MSSs within the computational budget and care less about the long-term convergence beyond the budget. Although RRD has weaker convergence guarantee in the limit, it is able to learn stable profiles quickly within the budget by controlling the degree of regularization, which fulfills the spirit of strategy exploration. If an absolute convergence is pursued, one can always switch RRD to DO at certain PSRO iteration after useful strategies have been quickly generated by RRD.

	a_2^1	a_2^2	a_2^3	...	a_2^{500}	...	a_2^{1000}
a_1^1	(0, 0)	(0, 0.011)	(0, 0)	...	(0. 0.01)	...	(0,0.005)
a_1^2	(0.011, 0)	(0.1, 0.1)	(0.1, 0.2)	...	...	...	(0,0.199)
a_1^3	(0, 0)	(0.2, 0.1)	(0.2, 0.2)	...	...	...	(0, 0)
...	...	...	...	...	...	...	...
a_1^{500}	(0.01, 0)	...	...	...	...	...	(0, 0)
...	...	...	...	...	...	...	...
a_1^{1000}	(0.005, 0)	(0.199, 0)	(0, 0)	...	(0, 0)	...	(100, 100)

Table 7 A game instance with a long NE path.