A PROOF OF CONVERGENCE THEOREM

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 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 infinite number of ϵ-NE in an empirical game given a specific ϵ, so the definition of closeness is associated with a specific ϵ-NE.

Next we prove that if an empirical game is 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 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 a profile σ with regret smaller than or equal to ϵ, i.e., an ϵ-NE. By setting $\epsilon = \lambda$, we prove the Theorem 1.

Proof. Since we have finite strategy space S, closeness with respect to certain σ is always reachable by training against an ϵ-NE at each iteration. Once the 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., 2021), 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. It is obvious that when $\epsilon = 0$, RRD reduces to DO as well as its theoretical convergence guarantee.

Note that similar results have been proved by McAleer et al. (2021) and Dinh 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 induces 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.

B EXTRA EXPERIMENTAL RESULTS

B.1 RESULTS FOR QRE BEING A 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 & Palfrey, 1995; 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 a 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.

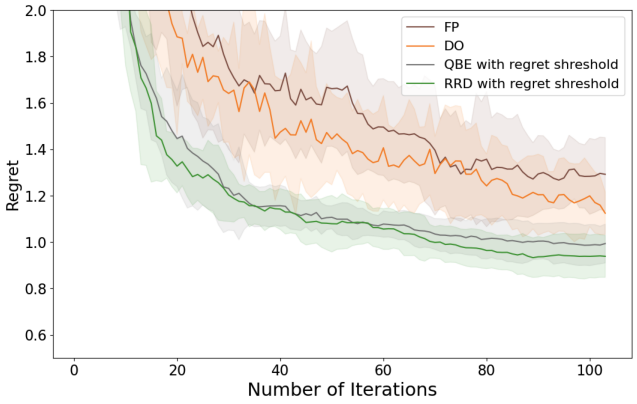
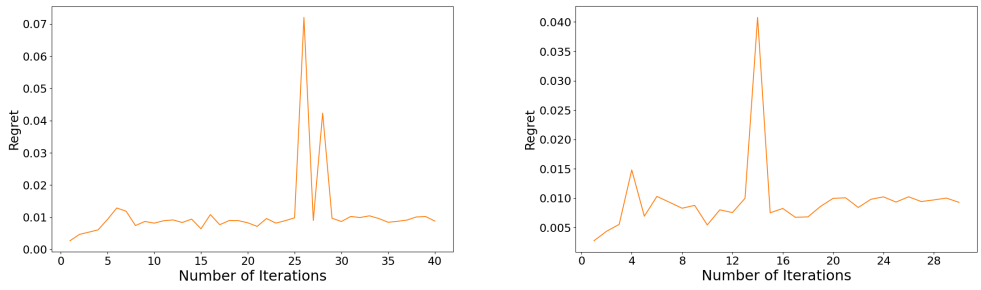


Figure 6: Learning performance with QRE.

Figure 6 shows the learning performance in Leduc poker with MSS QRE. 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 a MSS in PSRO.

B.2 Advantage over RD with Fixed Number of Iterations

Before designing RRD, we first carefully investigate the learning behavior of RD with a fixed number of iterations (RDFI) in PSRO. In other words, in every iteration of PSRO, we search towards an empirical NE with RD but fix the number of RD iterations. Figure 7 plots the regret of the profile given by different runs of RDFI with respect to the empirical game. From Figure 7, we observe that instability in regret (i.e., spikes) appear in nearly all runs of RDFI (despite we only show two of them). The values of spikes in the figures seem to be not large enough to affect the learning performance. This is because we select a large number of fixed RD iterations and seek for a convergence to NE. However, when applying a fixed number of RD iterations for regularization purpose, we find that the values of spikes can be very large and damage the quality of the best-response target. Therefore, we control the regret as in RRD rather than fixing the number of RD iterations to prevent RD from outputting an unstable profile.



(a) Run 1. (b) Run 2.

Figure 7: Spikes in regret curve of RDFI.

In Figure 8, we show the number of RD iterations 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 iterations

required to reach the regret threshold λ varies across PSRO iterations, which again emphasizes the need of adjusting the number of RD iterations dynamically rather than fixing the number of RD iterations. 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.

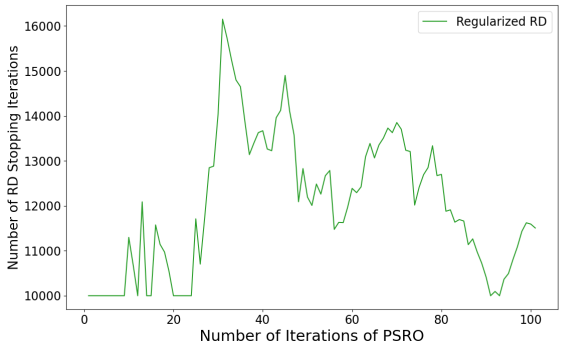


Figure 8: Number of iterations reaching λ.

B.3 Standard Deviation of Learning in Real-world Games

We report the standard deviation of last-iteration regret of Hex in Table 2. For RGS, we initialize PSRO with a uniform strategy. Since learning is performed in the matrix game, the regret at every iteration is deterministic given a fixed initial strategy. Therefore, no error bar is exhibited for RGS.

MSSs	**Hex**
FP	0.252
DO	0.153
RRD	0

Table 2: Standard deviation of regret curves

C Backward Profile Search

To handling games with a large number of players, Brinkman & Wellman (2016) employed a heuristic profile search with an incomplete payoff matrix, which selectively simulates payoffs of profiles in a game with NE convergence guarantee. Specifically, it first selects a set of subgames of the empirical game whose payoff matrices are complete, i.e., the payoffs of all profiles in the subgame have been simulated. Then NE of these subgames are proposed as candidate equilibria of the full game. For each candidate equilibrium, 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 a NE of the full game. Otherwise, deviation strategies are added to the subgames and the payoffs of new profiles are simulated. In the process, only payoff of pure strategy profile along the deviation path will be simulated instead of the whole payoff matrix. Therefore, a solution concept can be obtained by only simulating a small portion of profiles.

However, a simple combination of the original algorithm with PSRO could yield almost full payoff matrix being simulated since it is very likely that newly added strategies become beneficial deviations of profiles from previous subgames. To make it compatible with PSRO, we borrow its key spirit and propose a variant, called *backward profile search* (BPS). As shown in Algorithm 3, rather than starting from one complete subgame from previous iterations, BPS starts search from the singleton profile constituted by the newest strategies added to empirical game at the current iteration. Then BPS searches potential deviations back to strategies from previous PSRO iterations. Our motivation is

that new strategies generated in PSRO are more likely to constitute a NE compared to previous ones, enabling BPS to confirm a NE quickly. Once BPS confirms a NE, we apply RRD to the subgame that contains the NE rather than the whole empirical game payoff matrix, thus saving a moderate number of simulations. Note that in the case the profile proposed by RRD is a QRE of the subgame.

Algorithm 3 Backward Profile Search

Input: Empirical game $\mathcal{G}_{S\downarrow X}$ with incomplete payoff matrix.
Output: NE of $\mathcal{G}_{S\downarrow X}$ σ.

Initialize subgame Q with newly added strategy $s_{|X_i|}, \forall i \in N$.
while True, **do**
  $\sigma =$ NE_search(Q)
  $\bar{X} =$ deviation_strategies($\mathcal{G}_{S\downarrow X}, \sigma$)
  if $|\bar{X}| = 0$ **then**
    Break.
  else
    $s = \text{pop}(\bar{X})$ where $s_i = \text{argmax}_{s'} u_i(s', \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}), \forall i \in N$
  end if
  $Q = Q \bigcup s$
  Evaluate missing entries of payoff matrix of Q through simulation.
end while
return σ.

Figure 9 illustrates the mechanism of BPS at the third iteration of PSRO, in which each player has 4 strategies. The 4x4x4 cube in Figure 9 represents the payoff matrix of the current empirical game. Green cells represent the payoffs of evaluated profiles and white cells represent the payoffs of potential deviations. 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 from evaluating the newest profile (red cell) viewed as a subgame. Then BPS evaluates all payoffs of potential deviation profiles (white cells). Suppose the blue cell is a profile with the largest deviation payoff. We add the blue cell to the current subgame. Then we find NE of the subgame and search for potential deviation profile of the NE. Again, if the purple cell is a deviation profile, we add it to the subgame. We repeat this procedure until the NE of the subgame is confirmed.

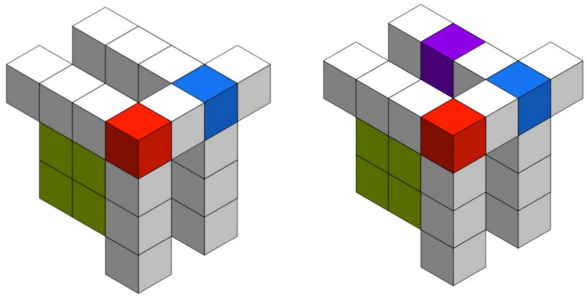


Figure 9: Illustration of BPS.

D Prior Explanations on Regularization

In the early study of strategy exploration, Schvartzman & Wellman (2009b) studied First-Price Sealed-Bid auction and found that adding noise to NE alters the path of equilibrium search by plotting the paths of DO and DO with noise in the strategy space. But it fails to explain how this noise contributes to the change of equilibrium search path. Under the PSRO framework, PRD is motivated by balancing overfitting to NE and exploration to the rest of the strategy space. Wang et al. (2019) presented similar motivation for combining FP and DO. Wright et al. (2019) found that fine-tuning best response against previous opponent largely accounts for the improved performance. Note that these work only described the motivations of their MSSs and showed the superior learning performance. Their

explanations are very general and do not elucidate the mechanism of regularization. In the same line of work, Balduzzi et al. (2019) introduced the concept *Gamescape* to refer to the scope of joint strategies covered by the exploration process to a given point. They employed this concept to characterize the effective diversity of an empirical game state, and proposed rectified Nash designed to increase diversity of the Gamescape. Compared to prior work, our new insight is based on quantified metric, i.e., regret w.r.t the full game, with which the affect of regularization is revealed.

E RESULTS FOR USING MRCP AS A MSS

We have observed the existence of strategy profiles with lower regret than NE in the empirical game. One natural question to ask is whether training against these stable profile targets benefits strategy exploration in real-world games, e.g., using MRCP as MSS. We hypothesize the existence of a connection between the learning performance and the regret of target profile, based on the observation that training against the stable profile target leads to better performance in our poker-game experiments.

To test the hypothesis, 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. 10a-10d show that with 3 out of 4 starting points, MRCP converges faster than DO. For the matrix game, Fig. 10e and 10f show the benefits of applying MRCP but the performance varies across different starting points.

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 a 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

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

To illustrate this concept, consider the symmetric zero-sum matrix game in Table 3. 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).

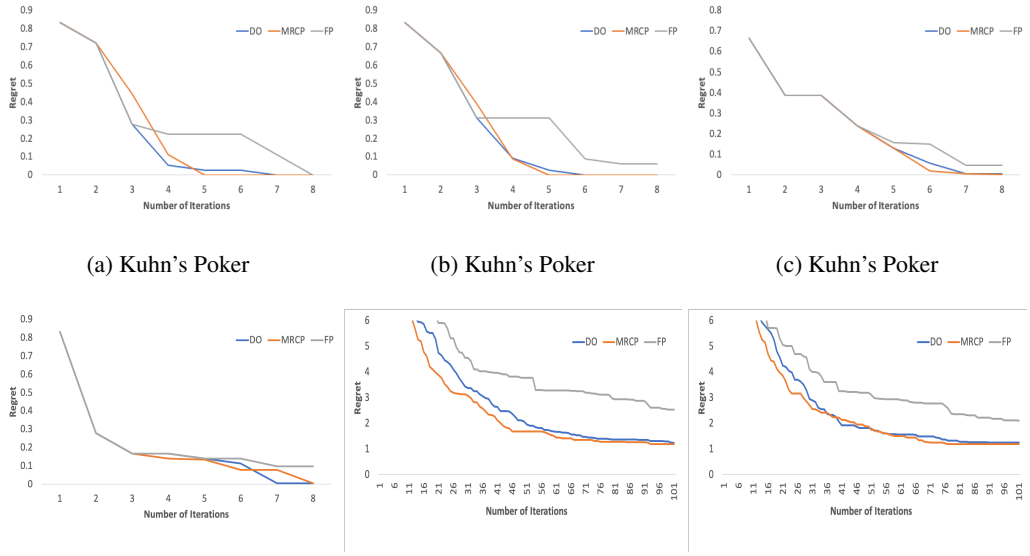
	a_2^1	a_2^2	a_2^3
a_1^1	(0, 0) [2]	(-1, 1) [6]	(-0.5, 0.5)
a_1^2	(1, -1) [6]	(0, 0) [10]	(-5, 5)
a_1^3	(0.5, -0.5)	(5, -5)	(0, 0)

Table 3: **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.

In Fig. 10, 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.

In the main paper, we claim that the lower-regret profile we select does not mean the better overall learning performance we obtain because the pursue of selecting an extremely stable profile as the training target may result in a slow update of the training target, yielding similar strategies to be added over PSRO iterations. We give an example for illustration and show the performance of using MRCP as a MSS in Appendix E. According to the definition, MRCP has the lowest regret with respect to 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 10: **MRCP as a MSS.** Y axis depicts MRCP-based regret.

true game, thus the profile closest to a true NE. From the experiments above, we find that learning with MRCP is feasible but does not accelerate learning compared to DO, which verifies our claim that selecting low-regret profile greedily may not slow down the update of the best response target and lead to overall good learning performance. Note that DO does not suffer this issue since the empirical NE would always change after adding a deviation strategy.

Now we illustrate why pursuing best response targets with extremely low regret may result in a slow learning using the matrix game shown in table 4.

	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 4: Game with long NE path.

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 $(0.005 * 0.5 + 0.199 * 0.5 - 0.011 * 0.25 - 0.1 * 0.25) * 2 = (0.102 - 0.02775) * 2 = 0.074252 = 0.1485$ (0.02 (regret of MRCP) $<$ 0.1485 (regret of RRD) $<$ 0.2 (regret of NE)).

In this example, best responding to the low full-game regret RRD profile 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 rather than staying similarly as MRCP. In summary, RRD is designed to propose a low regret profile that is close to the true NE but does not have regret as low as MRCP, in which case the best response target changes slowly yielding little overall performance improvement.

F No Free Lunch Theorem in Strategy Exploration

One general observation on game learning in a wide variety of different domains is that the performance of an algorithm can vary substantially across different games (Nudelman et al., 2004). Despite the demonstration that the state-of-the-art MSS maintains a satisfying performance over a variety of games, we believe no claim could be made that one MSS outperforms another MSS in every game. We describe this phenomenon as **No Free Lunch Theorem** in strategy exploration and give a descriptive reasoning to show its existence.

Denote the number of iterations that a MSS m converges as $I(m)$.

Theorem 2 (**No Free Lunch Theorem**). For any MSS m, there always exists an another MSS m' and a game $\mathcal{G} = (N, (S_i), (u_i))$ such that $I(m') < I(m)$ with the same starting point.

Proof. Denote as X_t and X'_t respectively the empirical games being developed by MSS m and m' at iteration t. Consider the iteration t at which X_t and X'_t differ and we denote the distinct strategy for each player as $s_t = \{s_1, s_2, ..., s_N\}$ and $s'_t = \{s'_1, s'_2, ..., s'_N\}$ under m and m', where s_i and s'_i could be none if player i's strategy sets given by m and m' are equal. One can always handcraft a game such that $s' = \{s'_1, s'_2, ..., s'_N\}$ is a NE of $\mathcal{G}$ (pick any existing strategy for player i if s'_i is none) while $X_t \bigcup s = \{s_1, s_2, ..., s_N\}$ does not contain a NE. □

The No Free Lunch Theorem has many implications for strategy exploration. For example, it indicates that the best MSS for all games does not exist. Therefore, the design of advanced MSSs should focus on either a specific type of game or average satisfying performance across games. The theorem also indicates that given any MSS there always exists a game in which the MSS should enumerate all strategies to find a NE. In a word, we hope to emphasize this general observation and believe this theorem would serve as a discipline in the future study of EGTA.

G Illustration of RRD

Consider the matrix game shown in Table 5 as an empirical game. (a_1^1, a_2^2) is a unique NE due to the dominance of a_2^2 and hence is the output of NE MSS. Notice that the utilities given by a_2^1 and a_2^2 are similar and the utility difference could be caused by estimation error. In this case, a uniform regularization would assign the same exploratory probability to a_2^1, a_2^3 and a_2^4. However, a_2^1 should be focused more than others and worth being explored for player 1. On the other hand, regularization using RD takes the relative importance of strategies into consideration and assigns exploratory probabilities based on utility information, which handles the homogeneous exploration issue.

	a_2^1	a_2^2	a_2^3	a_2^4
a_1^1	(9.9, 9.9)	(10, 10)	(-5, -5)	(-1, -1)

Table 5: Example of exploration with RD.

H EXPERIMENTAL PARAMETERS

We use OpenSpiel Lanctot et al. (2019) default parameter sets for experiments on Leduc and Kuhn's poker: 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 6. The poker games are asymmetric in the sense that one player always moves first.

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 6: DQN parameter

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 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 3-player Leduc poker, we experiment with $\lambda = 0.6$.