Deep Learning in Tree-Based Game Solving 5

Stephen McAleer

Outline of the next few lectures

- Deep learning in tree-based game solving 1
    - Deep learning recap
    - NFSP
    - Deep CFR
    - Policy gradient methods
- Deep learning in tree-based game solving 2
    - MCCFR
    - DREAM
    - ESCHER
    - NeuRD
- Deep learning in tree-based game solving 3
    - DeepNash for expert-level Stratego
- Deep learning in tree-based game solving 4
    - AlphaStar and OpenAI 5 for SOTA in video games
    - Double Oracle brief intro
- SOTA in double oracle algorithms
    - PSRO
    - XDO
    - SP-PSRO

A Taxonomy of Game-Theoretic RL

Lecture 1

- Counterfactual Regret Minimization (Zinkevich et al. 2007)
  - CFR: Zinkevich et al. 2007
  - MC-CFR: Lanctot et al. 2009
  - **Deep CFR: Brown et al. 2019**
  - DREAM: Steinberger et al. 2020
  - ESCHER: McAleer et al. 2022
- Policy Gradients
  - **Regret Policy Gradient (Srinivasan et al. 2018)**
  - OpenAI Five (OpenAI 2019)
  - Neural Replicator Dynamics (Hennes, Morrill, and Omidshafiei et al. 2020)
  - Actor Critic Hedge (Fu et al. 2022)
  - DeepNash for expert-level Stratego (Perolat, de Vylder, and Tuyls et al. 2022)
  - Magnetic Mirror Descent (Sokota et al. 2022)
- PSRO (McMahan et. al. 2003, Lanctot et al. 2017)
  - AlphaStar for expert-level Starcraft (Vinyals et al. 2019)
  - Pipeline PSRO (McAleer and Lanier et al. 2020)
  - α-PSRO (Muller et al. 2020)
  - XDO (McAleer et al. 2021)
  - Joint-PSRO (Marris et al. 2021)
  - Anytime PSRO (McAleer et al. 2022)
  - Self-Play PSRO (McAleer et al. 2022)
- **Neural Fictitious Self Play (Heinrich and Silver 2016)**

A Taxonomy of Game-Theoretic RL

- Counterfactual Regret Minimization (Zinkevich et al. 2007)
    - CFR: Zinkevich et al. 2007
    - **MC-CFR: Lanctot et al. 2009**
    - Deep CFR: Brown et al. 2019
    - **DREAM: Steinberger et al. 2020**
    - **ESCHER: McAleer et al. 2022**
- Policy Gradients
    - Regret Policy Gradient (Srinivasan et al. 2018)
    - OpenAI Five (OpenAI 2019)
    - **Neural Replicator Dynamics (Hennes, Morrill, and Omidshafiei et al. 2020)**
    - Actor Critic Hedge (Fu et al. 2022)
    - DeepNash for expert-level Stratego (Perolat, de Vylder, and Tuyls et al. 2022)
    - Magnetic Mirror Descent (Sokota et al. 2022)
- PSRO (McMahan et. al. 2003, Lanctot et al. 2017)
    - AlphaStar for expert-level Starcraft (Vinyals et al. 2019)
    - Pipeline PSRO (McAleer and Lanier et al. 2020)
    - α-PSRO (Muller et al. 2020)
    - XDO (McAleer et al. 2021)
    - Joint-PSRO (Marris et al. 2021)
    - Anytime PSRO (McAleer et al. 2022)
    - Self-Play PSRO (McAleer et al. 2022)
- Neural Fictitious Self Play (Heinrich and Silver 2016)

Lecture 2

A Taxonomy of Game-Theoretic RL

Lecture 3

- Counterfactual Regret Minimization (Zinkevich et al. 2007)
  - CFR: Zinkevich et al. 2007
  - MC-CFR: Lanctot et al. 2009
  - Deep CFR: Brown et al. 2019
  - DREAM: Steinberger et al. 2020
  - ESCHER: McAleer et al. 2022
- Policy Gradients
  - Regret Policy Gradient (Srinivasan et al. 2018)
  - OpenAI Five (OpenAI 2019)
  - Neural Replicator Dynamics (Hennes, Morrill, and Omidshafiei et al. 2020)
  - Actor Critic Hedge (Fu et al. 2022)
  - **DeepNash for expert-level Stratego (Perolat, de Vylder, and Tuyls et al. 2022)**
    - **From Poincaré Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization (Perolat et al. 2021)**
  - **Magnetic Mirror Descent (Sokota et al. 2022)**
- PSRO (McMahan et. al. 2003, Lanctot et al. 2017)
  - AlphaStar for expert-level Starcraft (Vinyals et al. 2019)
  - Pipeline PSRO (McAleer and Lanier et al. 2020)
  - α-PSRO (Muller et al. 2020)
  - XDO (McAleer et al. 2021)
  - Joint-PSRO (Marris et al. 2021)
  - Anytime PSRO (McAleer et al. 2022)
  - Self-Play PSRO (McAleer et al. 2022)
- Neural Fictitious Self Play (Heinrich and Silver 2016)

A Taxonomy of Game-Theoretic RL

- Counterfactual Regret Minimization (Zinkevich et al. 2007)
  - CFR: Zinkevich et al. 2007
  - MC-CFR: Lanctot et al. 2009
  - Deep CFR: Brown et al. 2019
  - DREAM: Steinberger et al. 2020
  - ESCHER: McAleer et al. 2022
- Policy Gradients
  - Regret Policy Gradient (Srinivasan et al. 2018)
  - **OpenAI Five (OpenAI 2019)**
  - Neural Replicator Dynamics (Hennes, Morrill, and Omidshafiei et al. 2020)
  - Actor Critic Hedge (Fu et al. 2022)
  - DeepNash for expert-level Stratego (Perolat, de Vylder, and Tuyls et al. 2022)
  - Magnetic Mirror Descent (Sokota et al. 2022)
- PSRO (McMahan et. al. 2003, Lanctot et al. 2017)
  - **AlphaStar for expert-level Starcraft (Vinyals et al. 2019)**
  - Pipeline PSRO (McAleer and Lanier et al. 2020)
  - α-PSRO (Muller et al. 2020)
  - XDO (McAleer et al. 2021)
  - Joint-PSRO (Marris et al. 2021)
  - Anytime PSRO (McAleer et al. 2022)
  - Self-Play PSRO (McAleer et al. 2022)
- Neural Fictitious Self Play (Heinrich and Silver 2016)

Lecture 4

A Taxonomy of Game-Theoretic RL

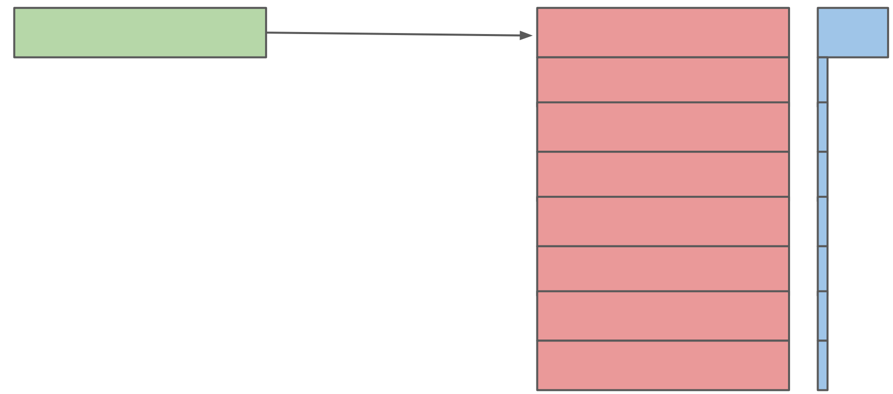
Lecture 5 (This Lecture)

- Counterfactual Regret Minimization (Zinkevich et al. 2007)
  - CFR: Zinkevich et al. 2007
  - MC-CFR: Lanctot et al. 2009
  - Deep CFR: Brown et al. 2019
  - DREAM: Steinberger et al. 2020
  - ESCHER: McAleer et al. 2022
- Policy Gradients
  - Regret Policy Gradient (Srinivasan et al. 2018)
  - OpenAI Five (OpenAI 2019)
  - Neural Replicator Dynamics (Hennes, Morrill, and Omidshafiei et al. 2020)
  - Actor Critic Hedge (Fu et al. 2022)
  - DeepNash for expert-level Stratego (Perolat, de Vylder, and Tuyls et al. 2022)
  - Magnetic Mirror Descent (Sokota et al. 2022)
- **PSRO (McMahan et. al. 2003, Lanctot et al. 2017)**
  - AlphaStar for expert-level Starcraft (Vinyals et al. 2019)
  - **Pipeline PSRO (McAleer and Lanier et al. 2020)**
  - α-PSRO (Muller et al. 2020)
  - **XDO (McAleer et al. 2021)**
  - Joint-PSRO (Marris et al. 2021)
  - **Anytime PSRO (McAleer et al. 2022)**
  - **Self-Play PSRO (McAleer et al. 2022)**
- Neural Fictitious Self Play (Heinrich and Silver 2016)

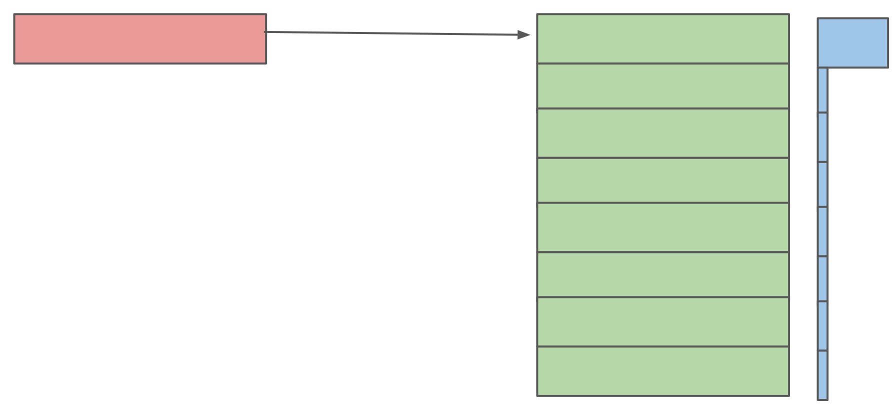
Self Play

- Both players learn best response to opponent's latest strategy
- Does not converge to a Nash equilibrium even in small games

Player 1 Best Responds to Player 2's Last Policy



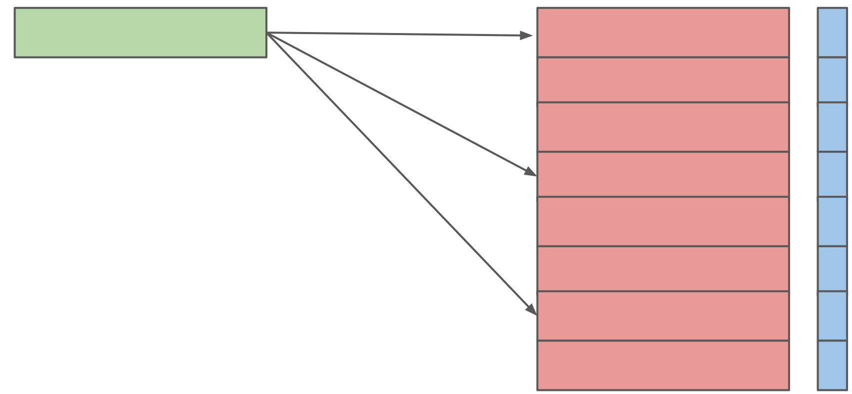
Player 2 Best Responds to Player 1's Last Policy



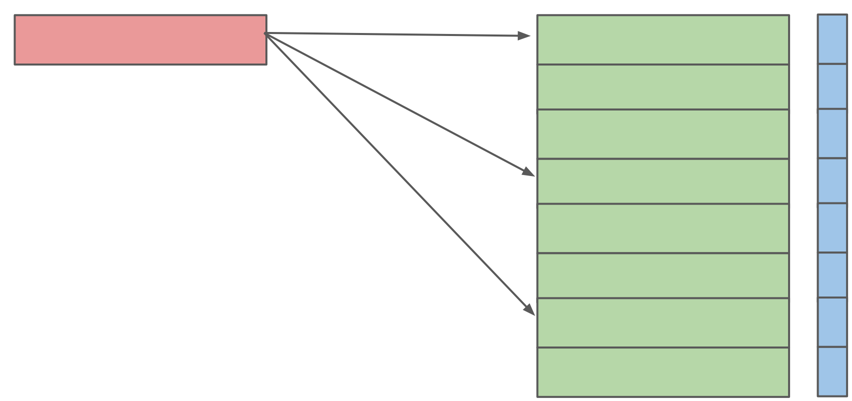
Fictitious Play

- Both players learn best response to opponent's average strategy
- Average strategy converges to a Nash equilibrium

Player 1 Best Responds to Player 2's Average Policy



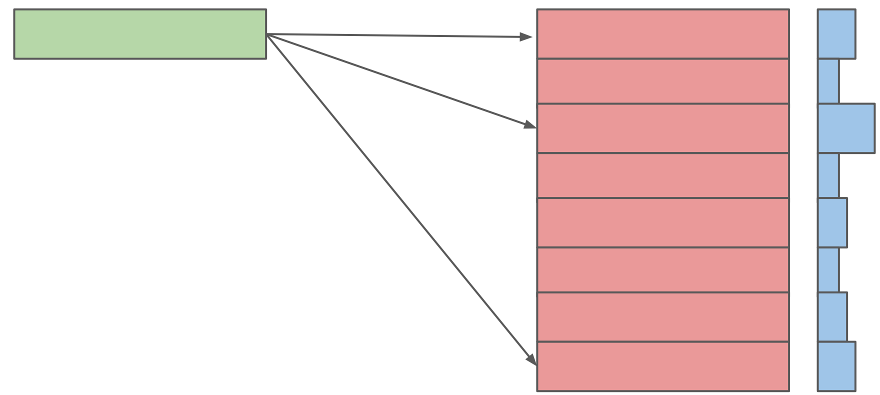
Player 2 Best Responds to Player 1's Average Policy



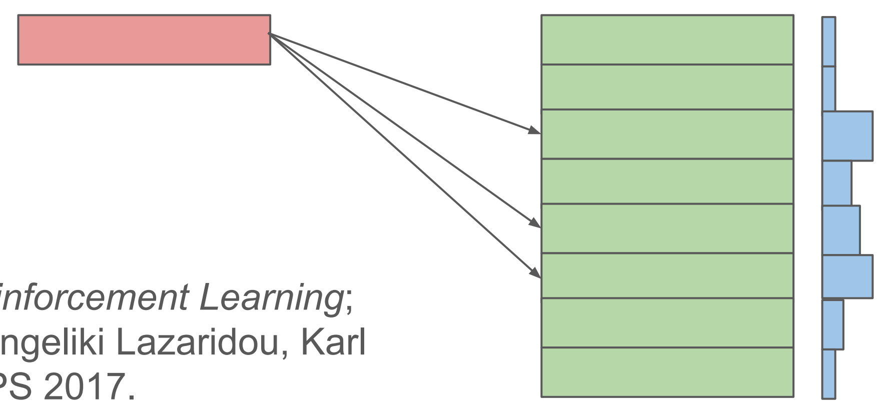
Policy Space Response Oracles (PSRO)

- Both players learn best response to opponent's meta-Nash
- Meta-Nash converges to a Nash equilibrium

Player 1 Best Responds to Player 2's Meta Nash



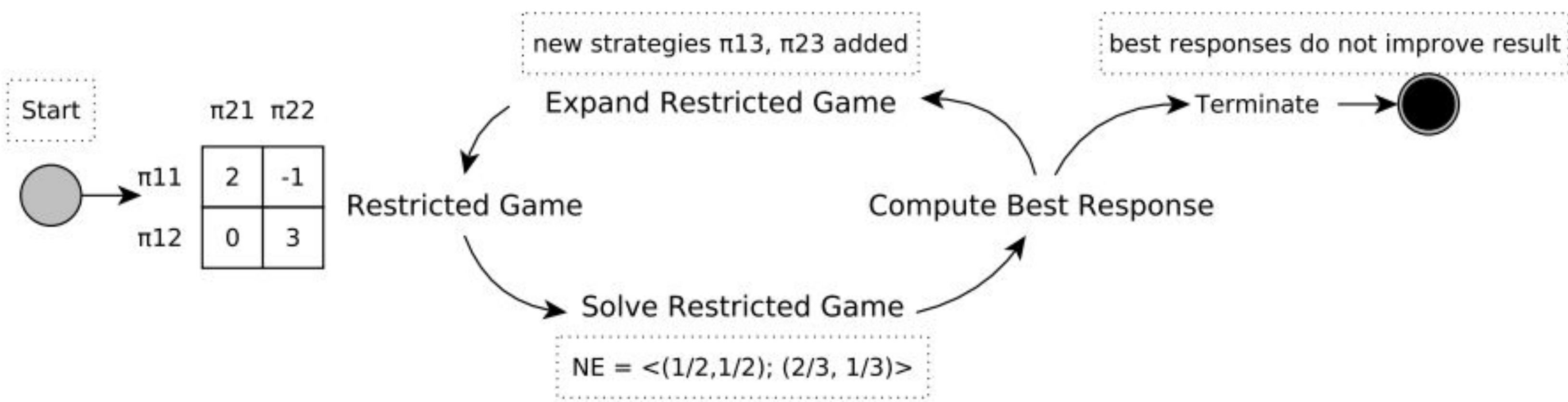
Player 2 Best Responds to Player 1's Meta Nash



A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning; Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, Thore Graepel. NIPS 2017.

PSRO

- Repeatedly add best responses to the meta-Nash to the population
- Meta-Nash is guaranteed to converge to Nash when enough strategies are added
- PSRO approximates best response through RL



PSRO Pros and Cons

- **Pros**
  - Can converge faster than NFSP, Deep CFR in certain games
  - Easy to use with any existing RL algorithm
  - Can handle continuous actions in practice
  - Has been used to achieve expert-level performance at Starcraft
- **Cons**
  - Sequential algorithm, requires training a new best response every iteration
  - Convergence guarantees on normal form of the game, exponential in # of infostates
  - Exploitability can increase from one iteration to the next
  - Strategies added every iteration are not optimal

Parallel PSRO

- DCH

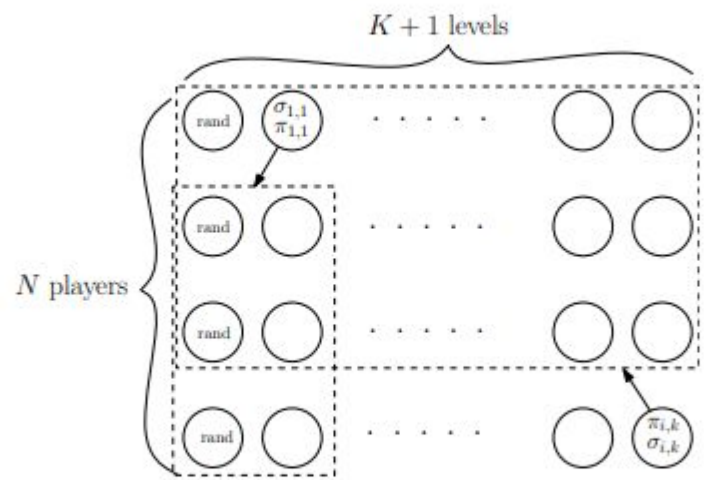


Figure 2: Overview of DCH

Parallel PSRO

- DCH
    - Need to know how many levels needed beforehand

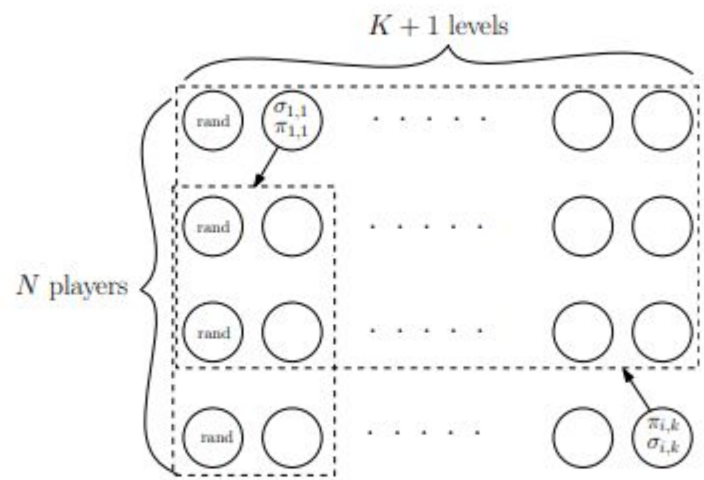


Figure 2: Overview of DCH

Parallel PSRO

- DCH
  - Need to know how many levels needed beforehand
  - Number of levels needed could be large

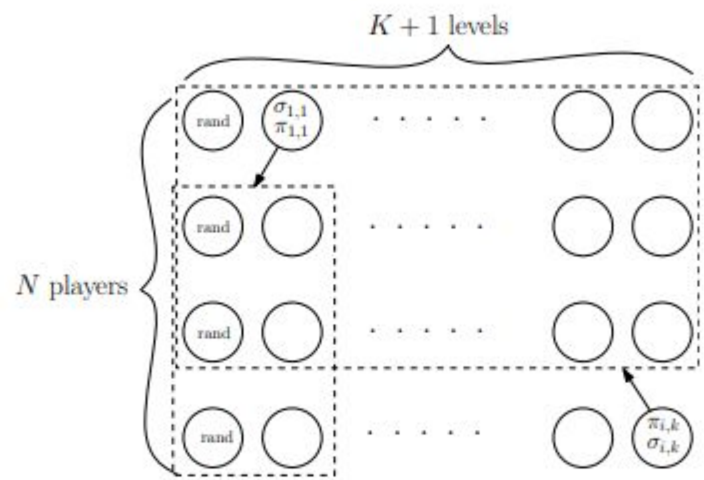


Figure 2: Overview of DCH

Parallel PSRO

- DCH
    - Need to know how many levels needed beforehand
    - Number of levels needed could be large
    - Randomness in best response causes ripple effect of instability

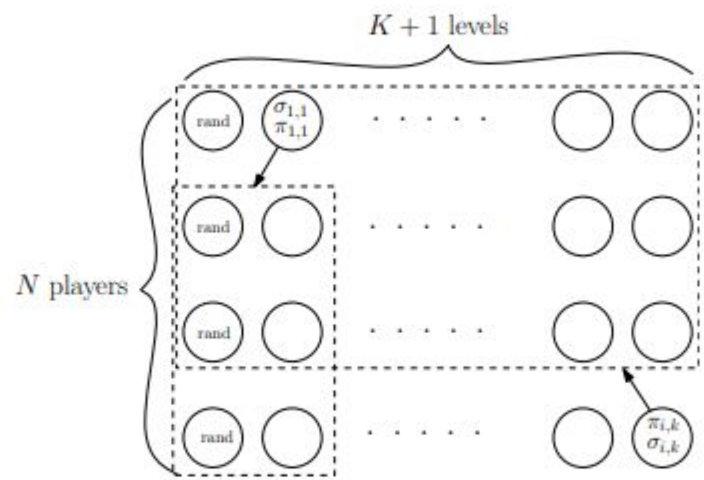


Figure 2: Overview of DCH

Parallel PSRO

- DCH
- Rectified PSRO

Algorithm 4 Response to rectified Nash (PSRO$_{\text{rN}}$)

input: population $\mathfrak{P}_1$
for $t = 1, \ldots, T$ **do**
  $\mathbf{p}_t \leftarrow$ Nash on $\mathbf{A}_{\mathfrak{P}_t}$
  for agent $\mathbf{v}_t$ with positive mass in $\mathbf{p}_t$ **do**
    $\mathbf{v}_{t+1} \leftarrow \text{oracle}\left(\mathbf{v}_t, \sum_{\mathbf{w}_i \in \mathfrak{P}_t} \mathbf{p}_t[i] \cdot \lfloor \phi_{\mathbf{w}_i}(\bullet) \rfloor_+\right)$
  end for
  $\mathfrak{P}_{t+1} \leftarrow \mathfrak{P}_t \cup \{\mathbf{v}_{t+1} : \text{updated above}\}$
end for
output: $\mathfrak{P}_{T+1}$

Parallel PSRO

- DCH
- Rectified PSRO
  - Not guaranteed to converge to Nash

Algorithm 4 Response to rectified Nash (PSRO$_{\text{rN}}$)

input: population $\mathfrak{P}_1$
for $t = 1, \ldots, T$ **do**
  $\mathbf{p}_t \leftarrow$ Nash on $\mathbf{A}_{\mathfrak{P}_t}$
  for agent $\mathbf{v}_t$ with positive mass in $\mathbf{p}_t$ **do**
    $\mathbf{v}_{t+1} \leftarrow \text{oracle}\left(\mathbf{v}_t, \sum_{\mathbf{w}_i \in \mathfrak{P}_t} \mathbf{p}_t[i] \cdot \lfloor \phi_{\mathbf{w}_i}(\bullet) \rfloor_+\right)$
  end for
  $\mathfrak{P}_{t+1} \leftarrow \mathfrak{P}_t \cup \{\mathbf{v}_{t+1} : \text{updated above}\}$
end for
output: $\mathfrak{P}_{T+1}$

$$\begin{bmatrix} 0 & -1 & 1 & -\frac{2}{5} \\ 1 & 0 & -1 & -\frac{2}{5} \\ -1 & 1 & 0 & -\frac{2}{5} \\ \frac{2}{5} & \frac{2}{5} & \frac{2}{5} & 0 \end{bmatrix}$$

How to Parallelize PSRO?

- DCH
- Rectified PSRO
- AlphaStar
  - Main agents, main exploiter agents, league exploiter agents

How to Parallelize PSRO?

- DCH
- Rectified PSRO
- AlphaStar
  - Main agents, main exploiter agents, league exploiter agents
  - Not proven to converge to Nash

How to Parallelize PSRO?

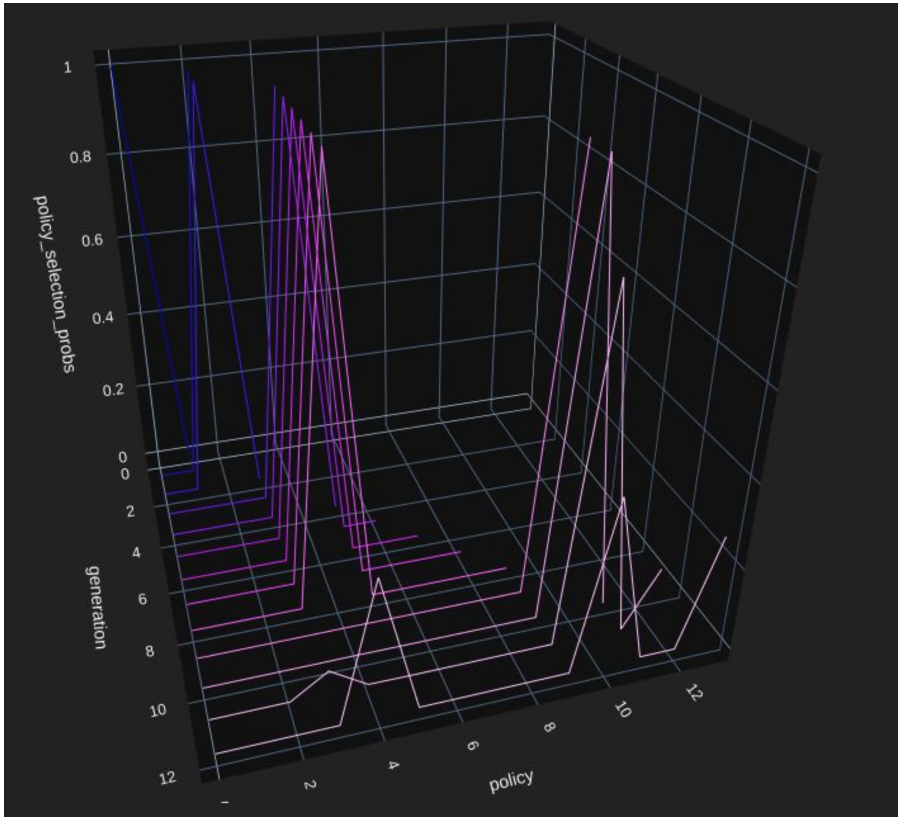
- DCH
- Rectified PSRO
- AlphaStar
  - Main agents, main exploiter agents, league exploiter agents
  - Not proven to converge to Nash
  - Could be difficult to replicate

How to Parallelize PSRO?

- DCH
- Rectified PSRO
- AlphaStar
  - Main agents, main exploiter agents, league exploiter agents
  - Not proven to converge to Nash
  - Could be difficult to replicate
  - Empirically (our implementation) can fail on normal form games

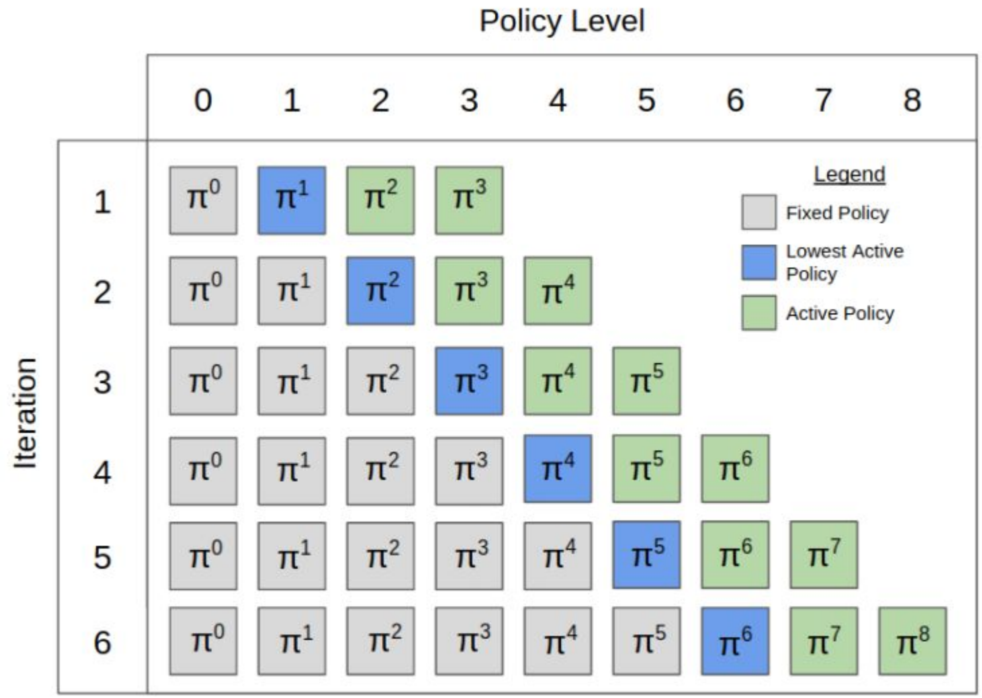
Parallel

- DCH
- Rectified PSRO
- AlphaStar
- Naive Parallel PSRO
  - Have each additional worker play against same meta-Nash distribution



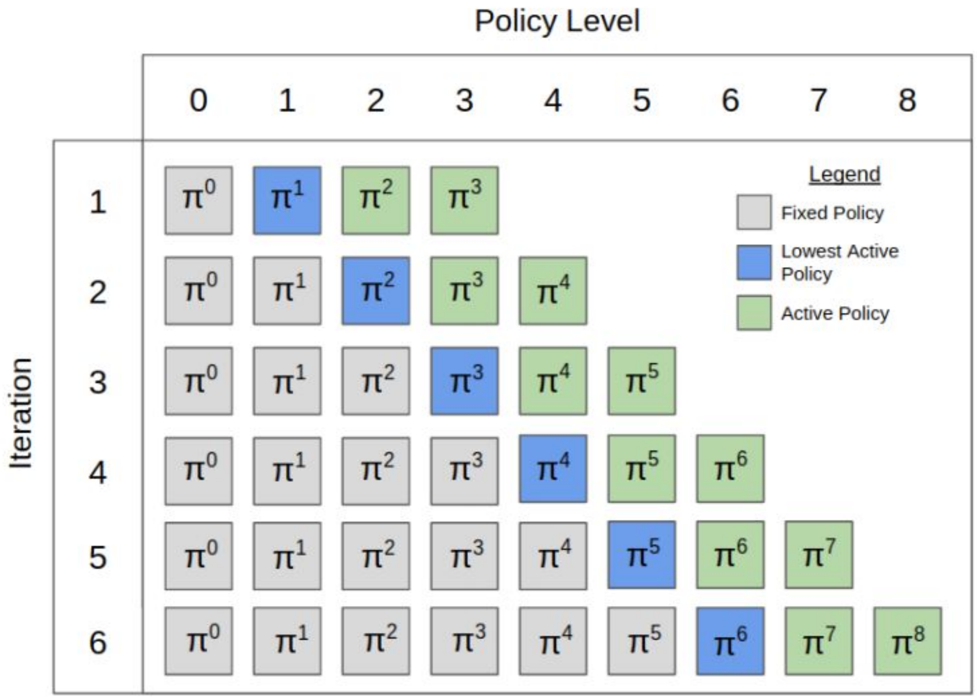
Pipeline PSRO

- Fixed and active policies



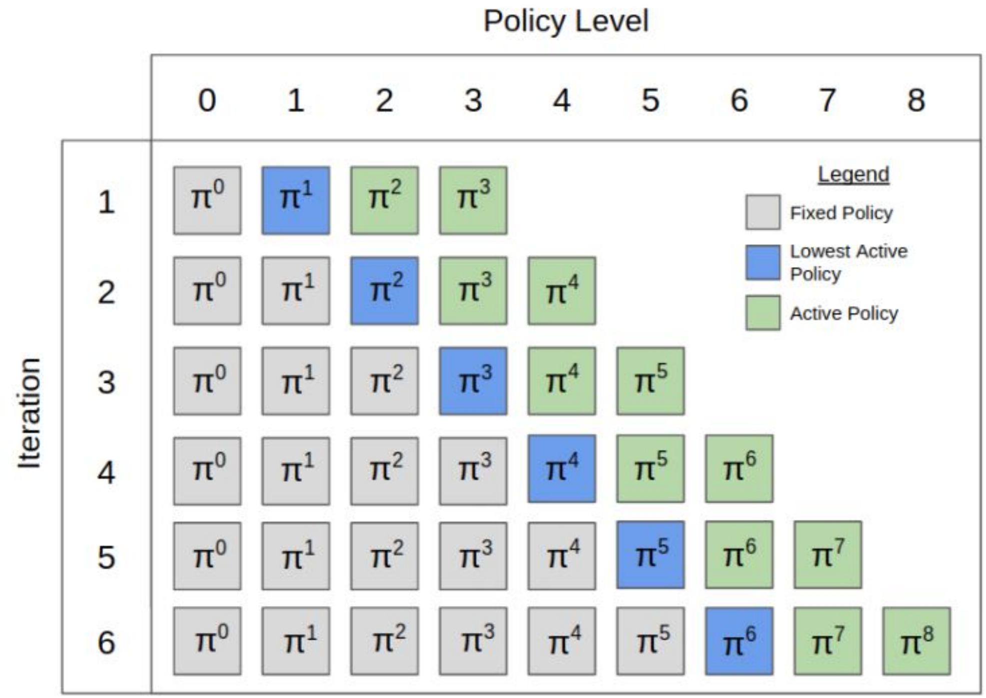
Pipeline PSRO

- Fixed and active policies
- Each active policy plays against meta-Nash of policies below it



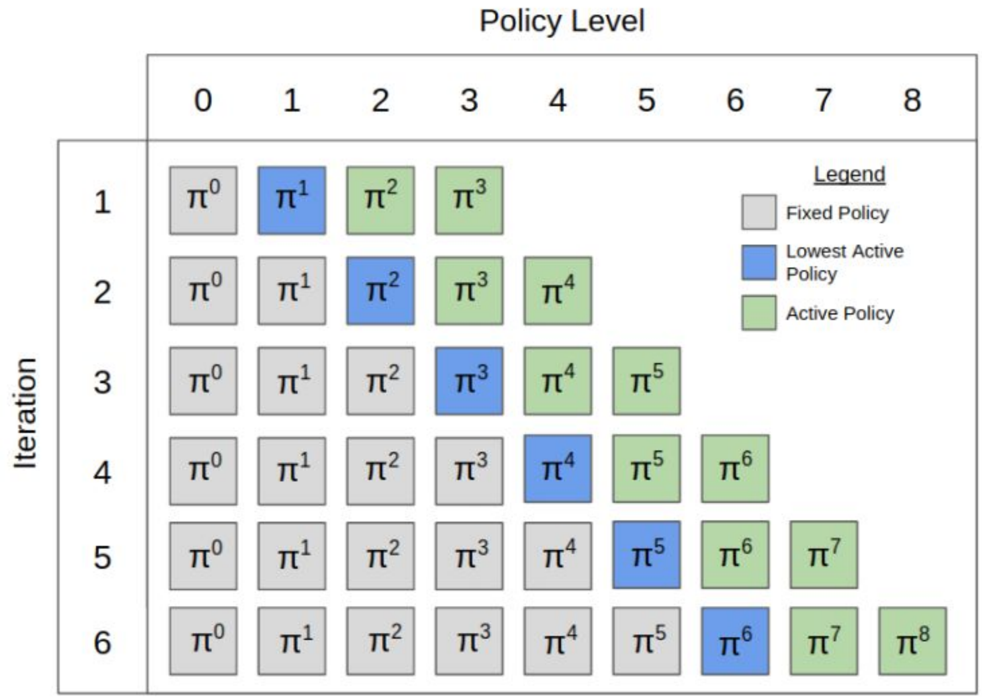
Pipeline PSRO

- Fixed and active policies
- Each active policy plays against meta-Nash of policies below it
- Once lowest active policy plateaus, it becomes fixed and a new policy is added

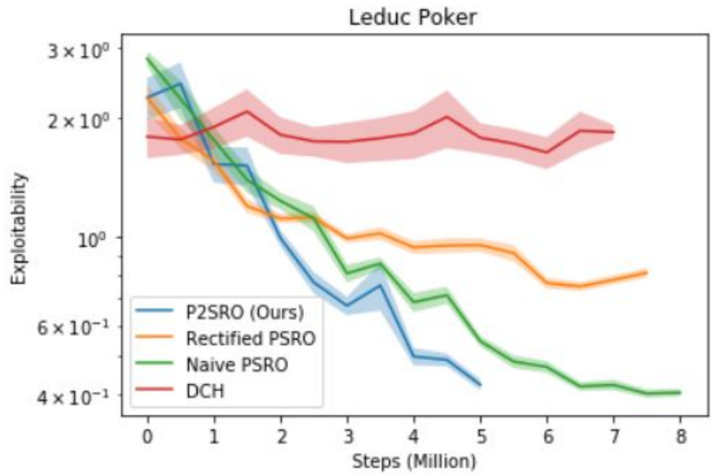


Pipeline PSRO

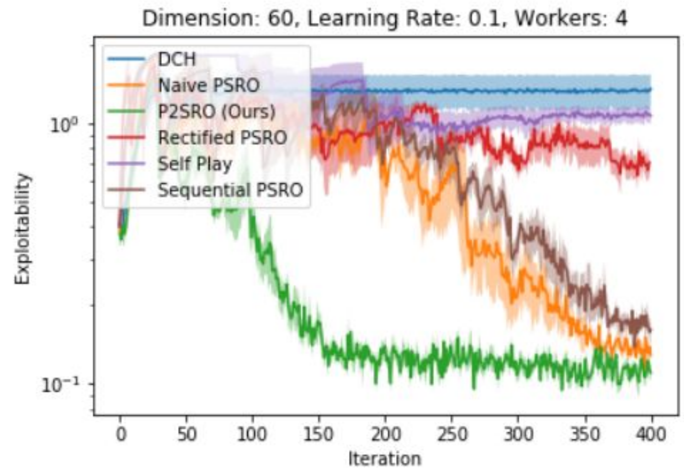
- Fixed and active policies
- Each active policy plays against meta-Nash of policies below it
- Once lowest active policy plateaus, it becomes fixed and a new policy is added
- Inherits same convergence guarantees as PSRO



Pipeline PSRO: Results



(a) Leduc poker

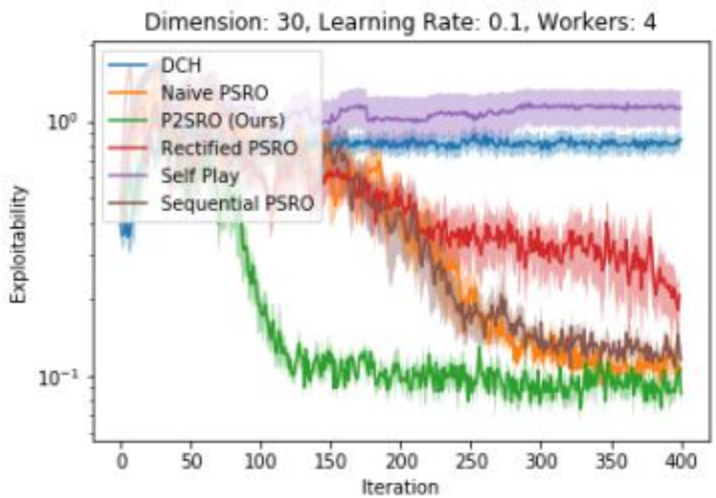


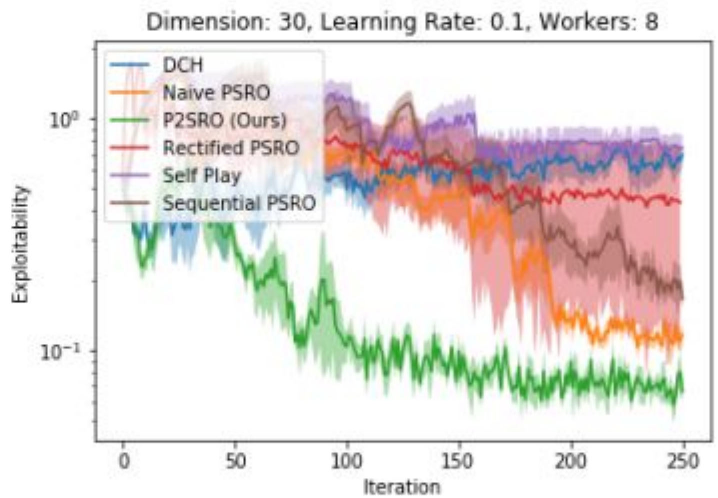
(b) Random Symmetric Normal Form Games

Figure 2: Exploitability of Algorithms on Leduc poker and Random Symmetric Normal Form Games

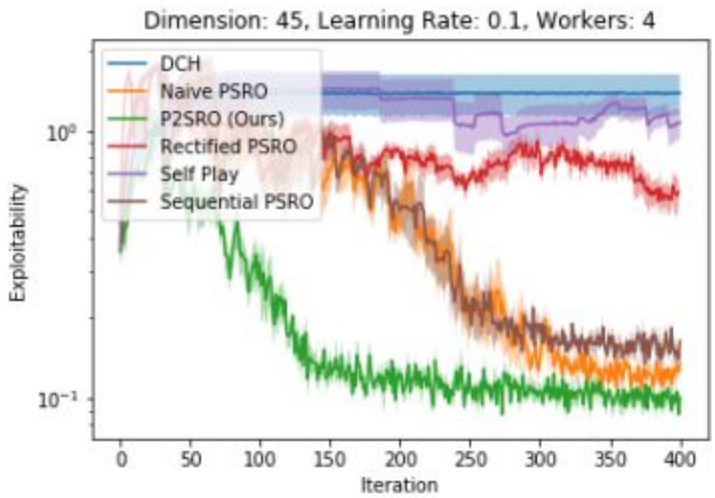
$$\pi' = r\mathrm{BR}(\hat{\pi}) + (1 - r)\pi$$

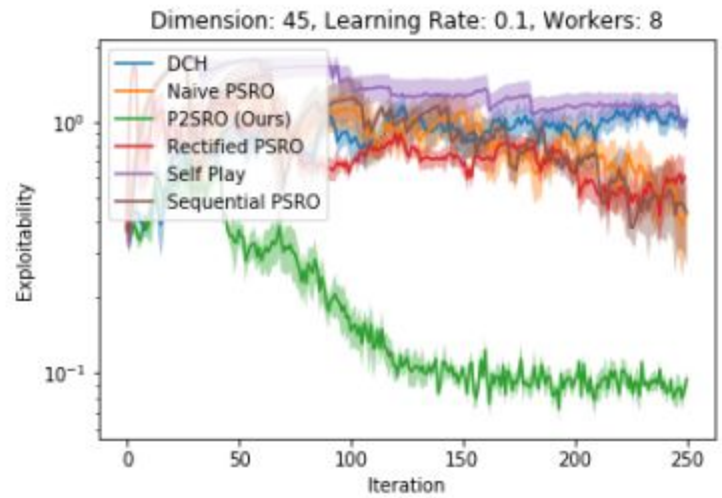
Pipeline PSRO: Results



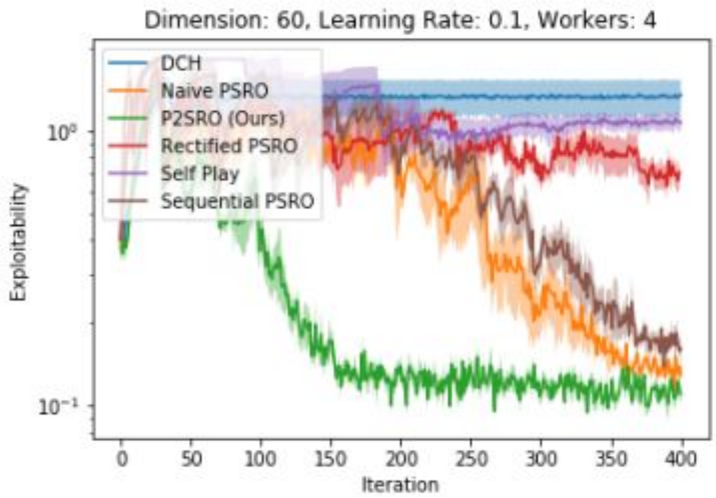


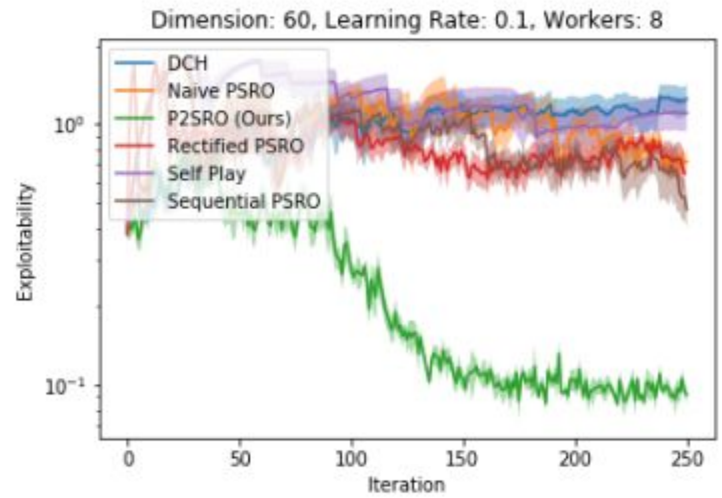
Pipeline PSRO: Results



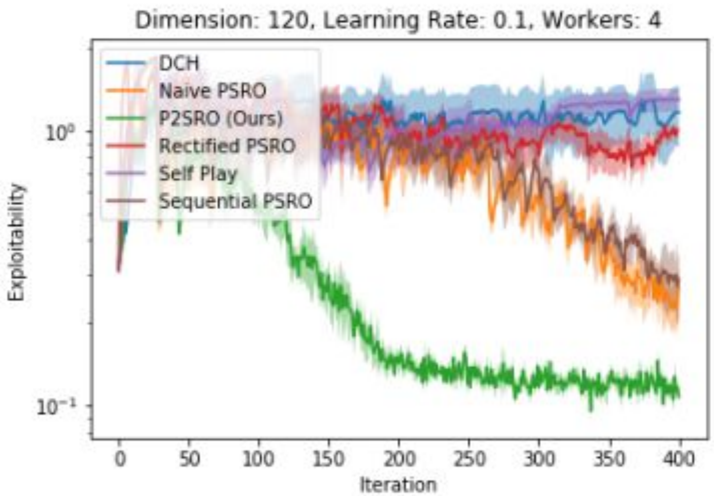


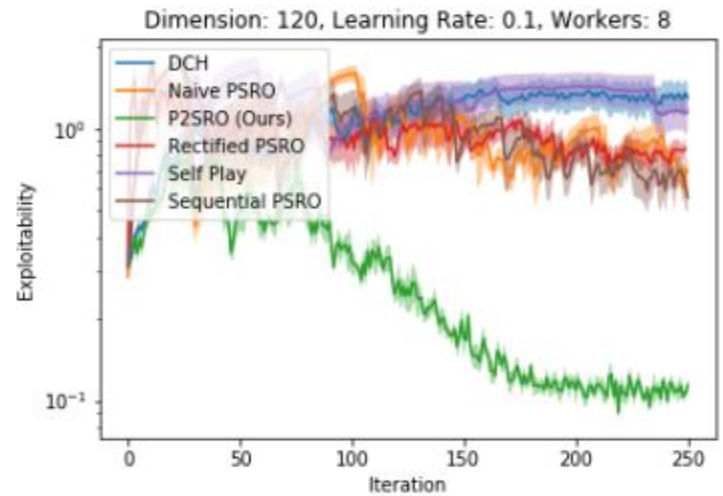
Pipeline PSRO: Results





Pipeline PSRO: Results





Pipeline PSRO Results: Barrage Stratego

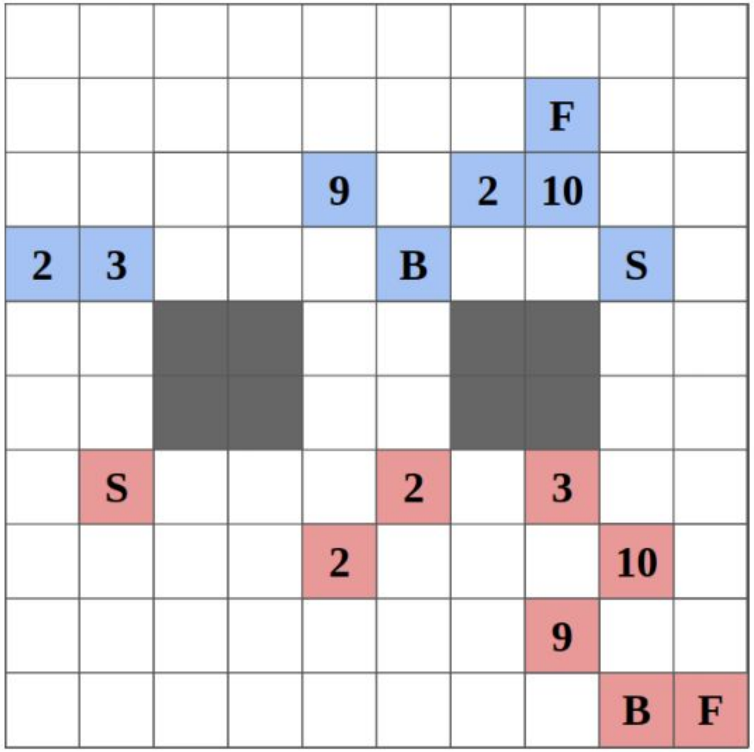


Table 1: P2SRO Results vs. Existing Bots

Name	P2SRO Win Rate vs. Bot
Asmodeus	81%
Celsius	70%
Vixen	69%
Celsius1.1	65%
All Bots Average	**71%**

PSRO Bad Case

- Because PSRO is a normal form algorithm, guarantees exist only in the number of normal form strategies
- But could need exponential number of normal form strategies to support Nash
- Can construct games where PSRO empirically expands all normal-form pure strategies

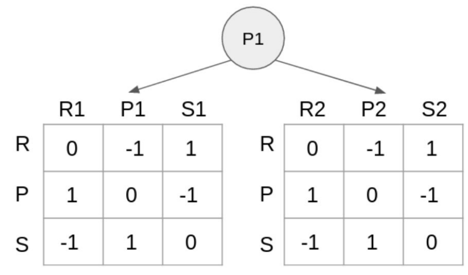
	H	T
H	1	-1
T	-1	1

↓

	H	T
H	1	-1
T	-1	1

PSRO Bad Case

- Because PSRO is a normal form algorithm, guarantees exist only in the number of normal form strategies
- Can construct games where PSRO empirically expands all normal-form pure strategies

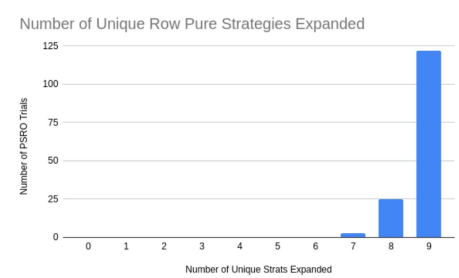


(a)

	R1	P1	S1	R2	P2	S2
RR	0	-1	1	0	-1	1
RP	0	-1	1	1	0	-1
RS	0	-1	1	-1	1	0
PR	1	0	-1	0	-1	1
PP	1	0	-1	1	0	-1
PS	1	0	-1	-1	1	0
SR	-1	1	0	0	-1	1
SP	-1	1	0	1	0	-1
SS	-1	1	0	-1	1	0

(b)

Figure 5. (a) Player 1 first chooses which RPS game both players play. Both players know which RPS game they are playing. Then both players simultaneously make their move. (b) The normal form game. Player 2 has 9 pure strategies.



Extensive-Form Double Oracle (XDO) Idea

- Instead of mixing over normal form strategies at the root of the game, allow mixing at every infostate
- Now only need HH and TT

	[illegible]	T
H	1	-1
T	-1	1

	H	T
H	1	-1
T	-1	1

(N)XDO Algorithm

- Same as PSRO, but meta-Nash is computed in extensive form of the game
- Restricted game is created by restricting the actions to be choosing a best response from the population
- This restricted game is solved via NFSP or CFR to get meta-Nash
- Linear convergence instead of exponential

Algorithm 1 XDO

Input: initial population Π^0
repeat
  Define restricted game for Π^t via equation (1)
  Get ϵ-NE policy π^{r*} of restricted game
  Find $\mathbb{BR}_i(\pi^{r*}_{-i})$ for $i \in \{1,2\}$
  if $v_i(\mathbb{BR}_i(\pi^{r*}_{-i}), \pi^{r*}_{-i}) \leqslant v_i(\pi^{r*}) + \epsilon$ for both i **then**
    Terminate
  $\Pi^{t+1}_i = \Pi^t_i \cup \mathbb{BR}_i(\pi^{r*}_{-i})$ for $i \in \{1,2\}$

$$\mathcal{A}^r_i(s_i) = \{a \in \mathcal{A}_i(s_i) : \exists \pi_i \in \Pi^t \text{ s.t. } \pi_i(s_i, a) = 1\} \quad (1)$$

Algorithm 2 NXDO

Input: initial population Π^0
repeat
  Define restricted game for Π^t via eq. (2)
  Get ϵ-NE policy π^{r*} of restricted game via NFSP
  Find $\mathbb{BR}_i(\pi^{r*}_{-i})$ for $i \in \{1,2\}$ via DRL
  $\Pi^{t+1}_i = \Pi^t_i \cup \mathbb{BR}_i(\pi^{r*}_{-i})$ for $i \in \{1,2\}$

XDO: A Double Oracle Algorithm for Extensive-Form Games; Stephen McAleer, John Lanier, Kevin Wang, Pierre Baldi, Roy Fox. NeurIPS 2021.

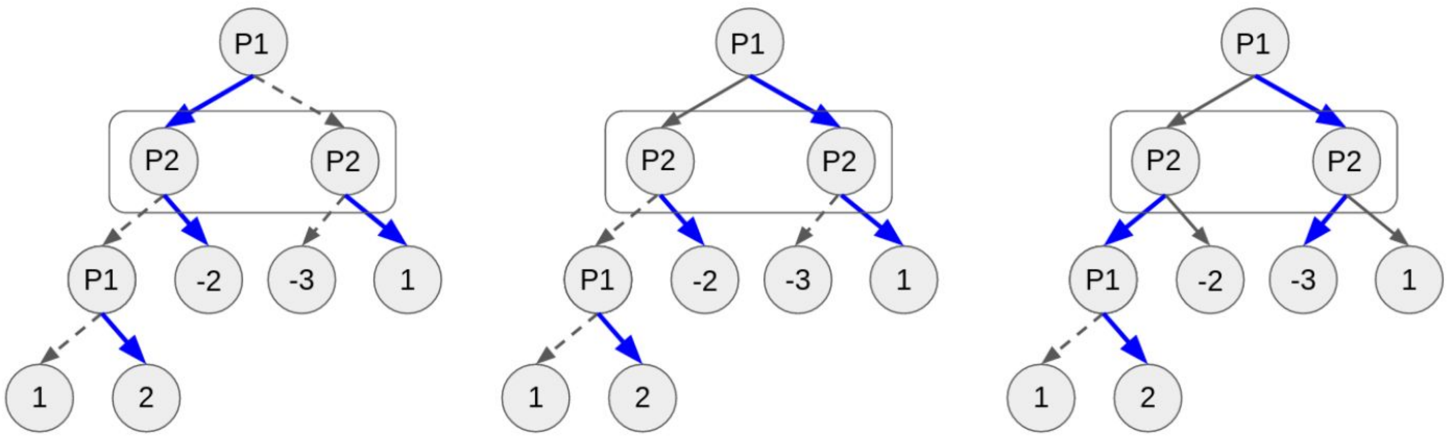
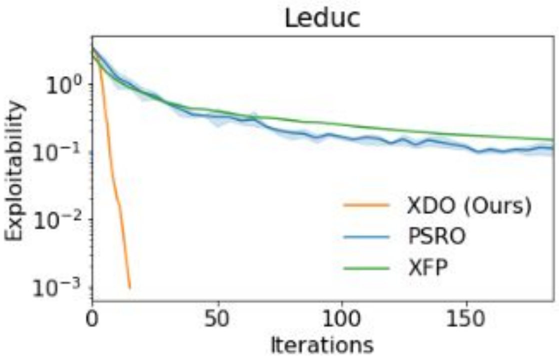
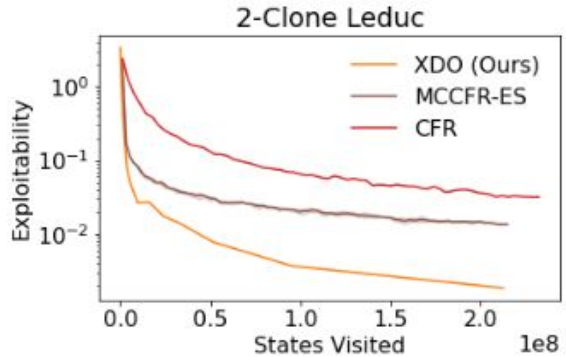


Figure 1. Three iterations of XDO (left to right). In these extensive-form game diagrams, player 1 (P1) plays at the root, then P2 plays without knowing P1's action, and if both played Left P1 plays another action. Actions in the restricted game are solid, vs. dashed outside the restricted game. Meta-NE actions are blue, vs. black not in the meta-NE. BR actions are thick, vs. thin for non-BR actions.

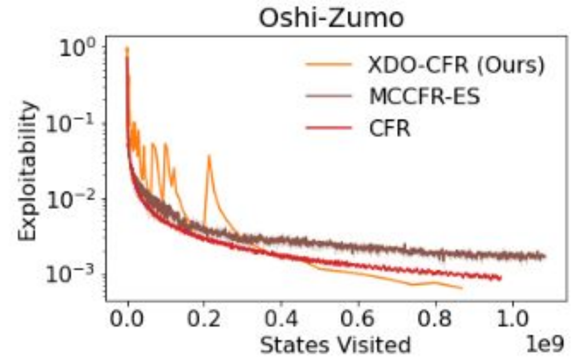
XDO Results



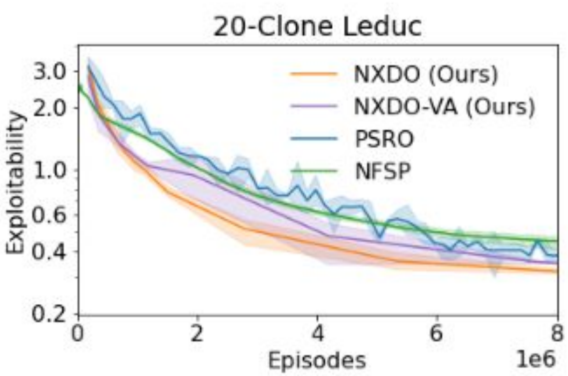
(a)



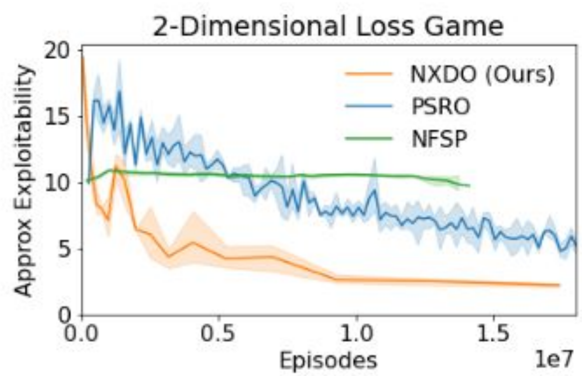
(b)



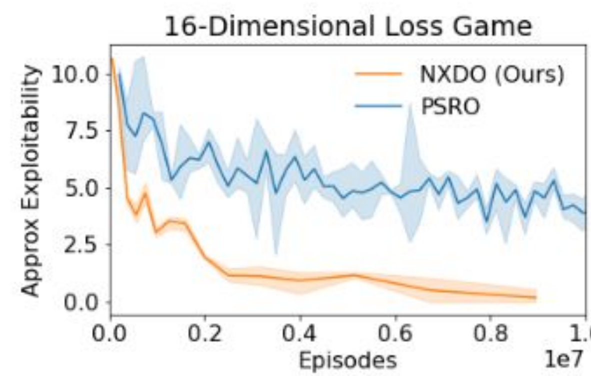
(c)



(a)



(b)



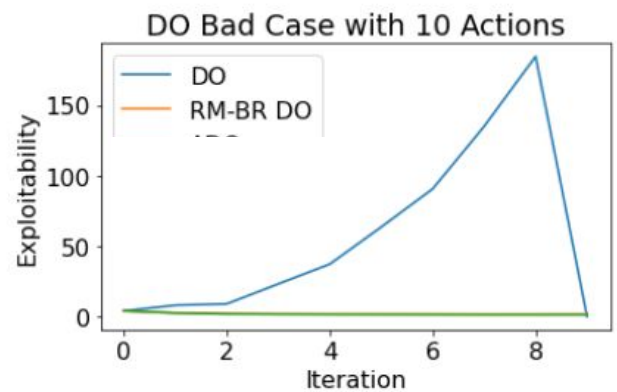
(c)

PSRO Can Increase Exploitability

100%

- PSRO is guaranteed to converge to a Nash if you run for enough iterations.
- But if you stop before convergence, the exploitability can be arbitrarily high
- This is because NE of restricted game is not least-exploitable distribution over population

		R	P	S
	R	0	-1	1
100%	P	1	0	-2
	S	-1	2	0



Least-Exploitable Restricted Distribution

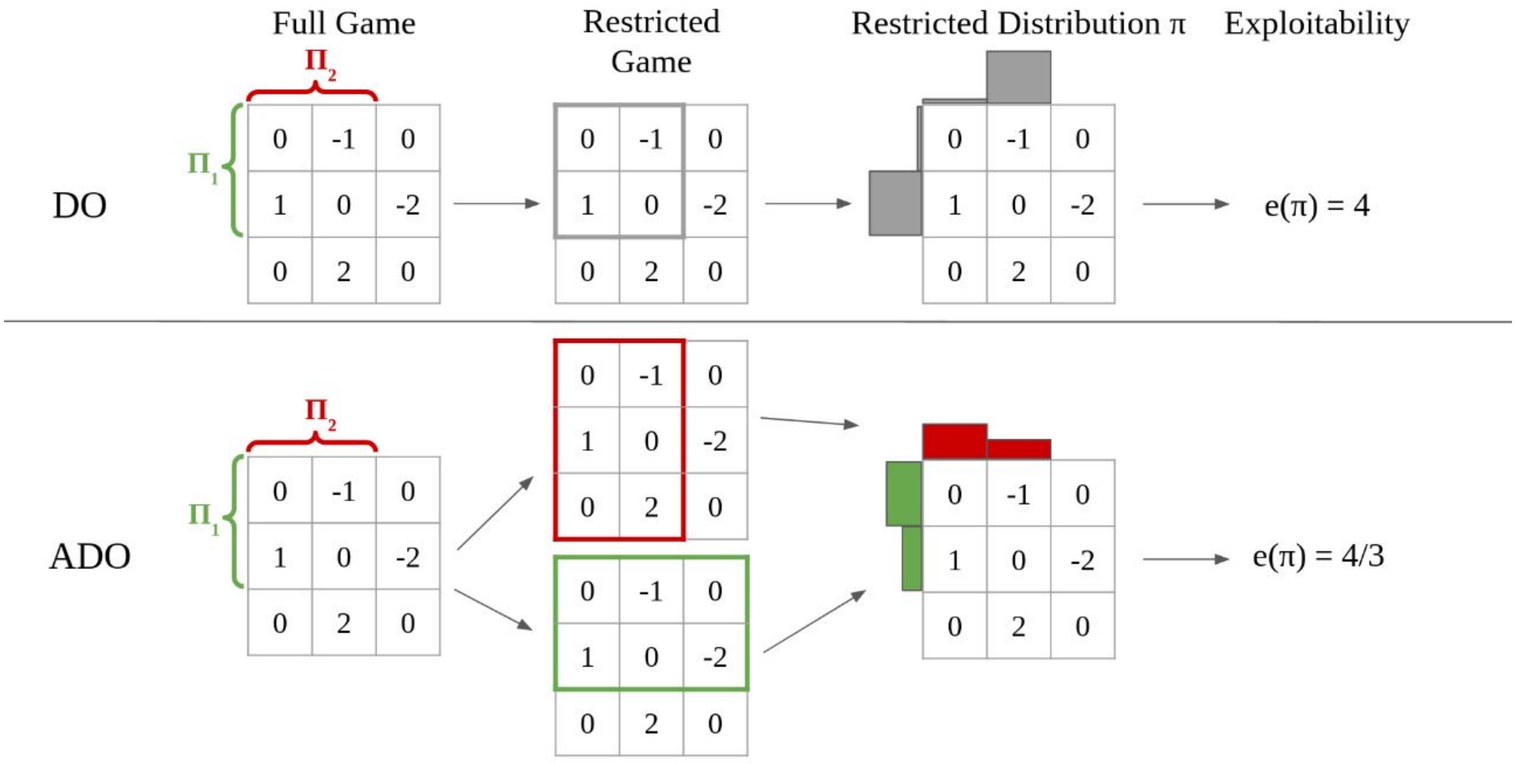
- Instead of computing meta-NE on restricted game, define new restricted game where opponent is unrestricted
- NE of this will be least-exploitable distribution over population
- Now, adding population members can only decrease least-exploitable distribution

$$\max_{\pi_i \in \Pi_i} \min_{\pi_{-i}} u_i(\pi_i, \pi_{-i}).$$

		R	P	S
50%	**R**	0	-1	1
50%	**P**	1	0	-2
	S	-1	2	0

	50%	50%	
	R	**P**	**S**
R	0	-1	1
P	1	0	-2
S	-1	2	0

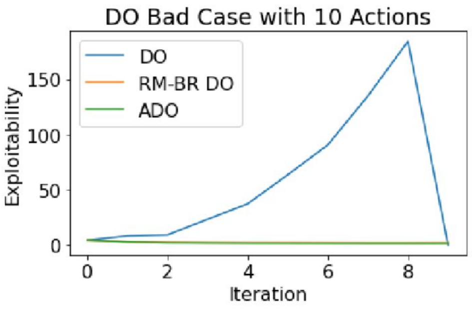
Anytime Double Oracle (ADO)

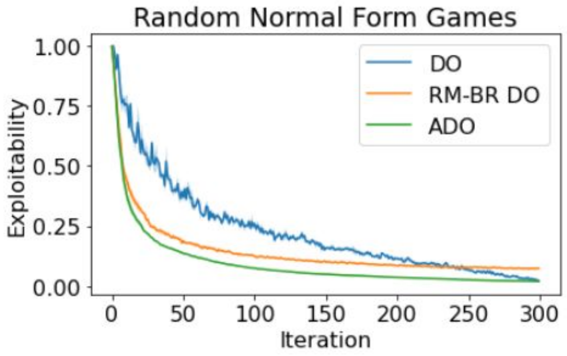


Anytime PSRO for Two-Player Zero-Sum Games; Stephen McAleer, Kevin Wang, Marc Lanctot, John Lanier, Pierre Baldi, Roy Fox. AAAI RLG Workshop 2022.

ADO Results

- Avoids DO counterexample and doesn't increase exploitability
- On random normal form games we achieve significantly lower exploitability every iteration



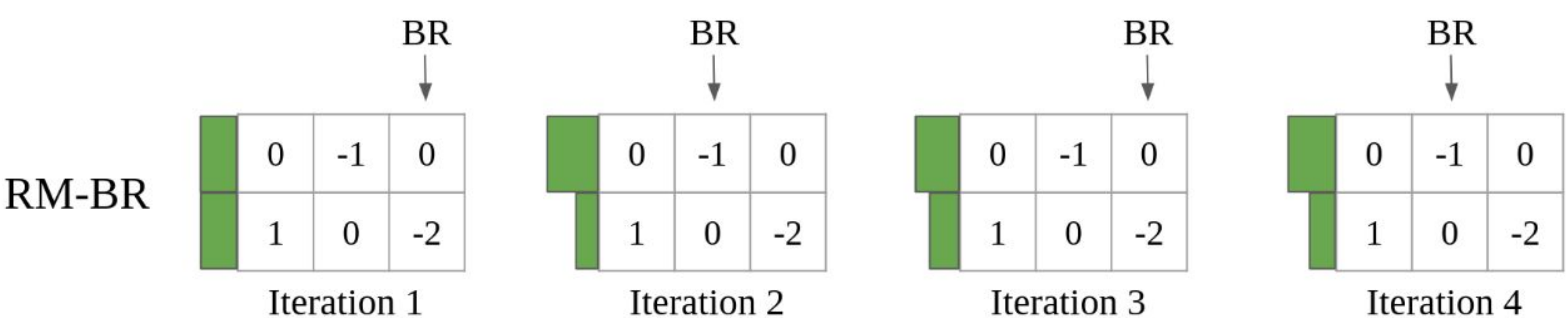


Regret-Minimizing against a BR Double Oracle (RM-BR DO)

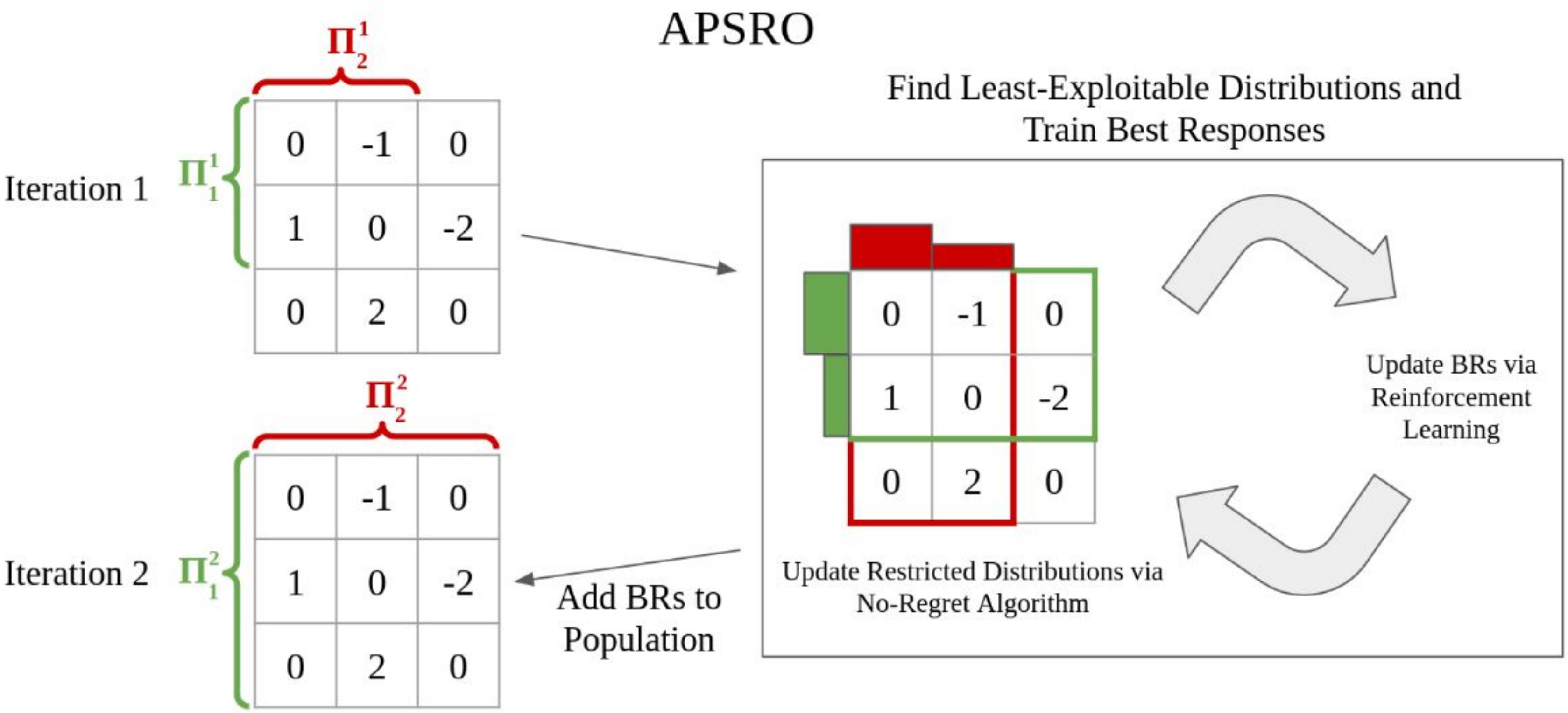
- Regret minimization against a BR will also converge to a Nash
- Can incorporate into double oracle algorithm to build foundation for next algorithm
- Will converge to ε-Nash and not increase exploitability

Algorithm 5: RM-BR DO

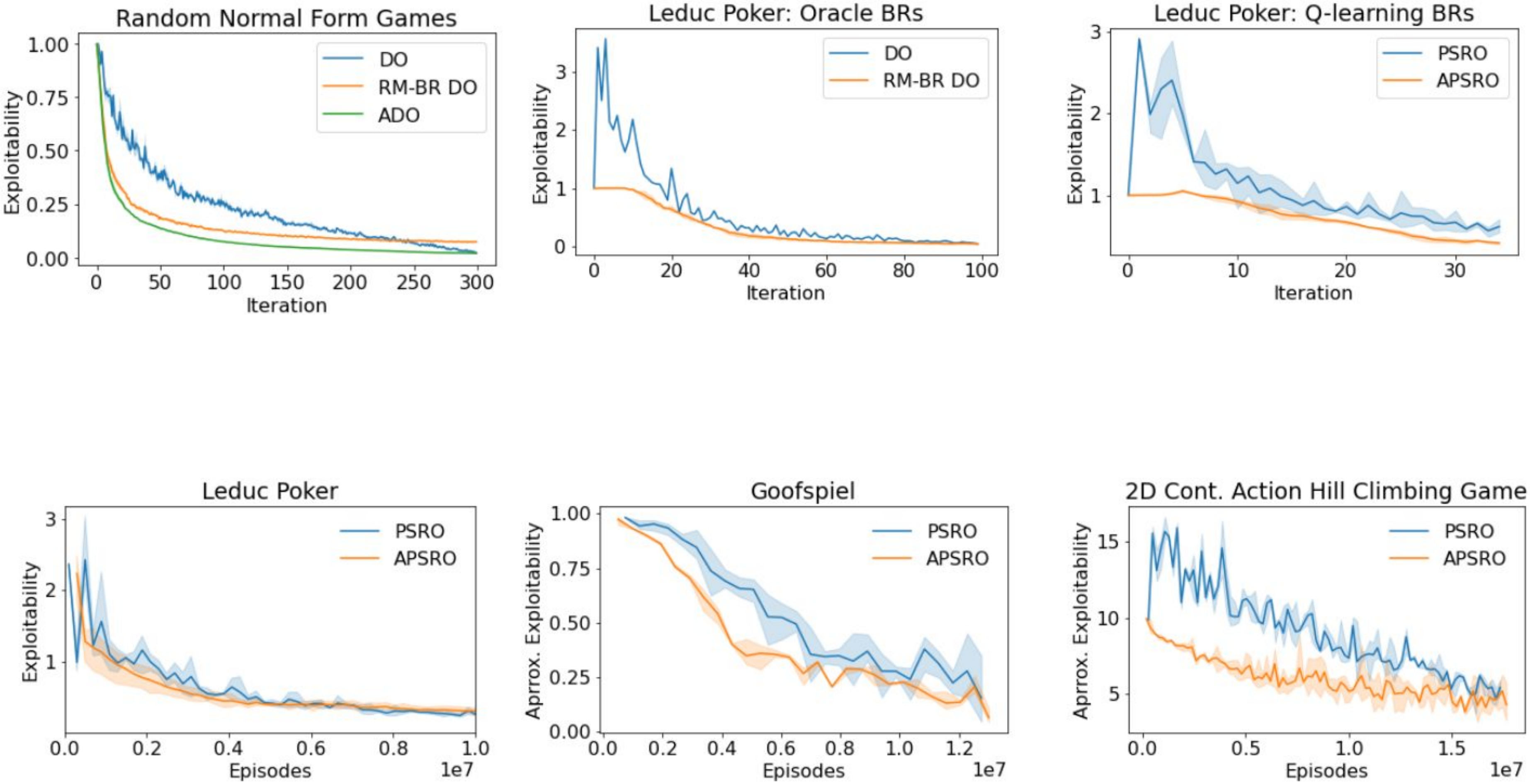
Result: Approximate Nash Equilibrium
Input: initial population Π^0
while *Not terminated* **do**
  Get meta-distribution π^r via RM-BR
  Find $\mathbb{BR}_i(\pi^r_{-i})$ for $i \in \{1, 2\}$
  $\Pi_i^{t+1} = \Pi_i^t \cup \mathbb{BR}_i(\pi^r_{-i})$ for $i \in \{1, 2\}$
end



Anytime PSRO (APSRO)



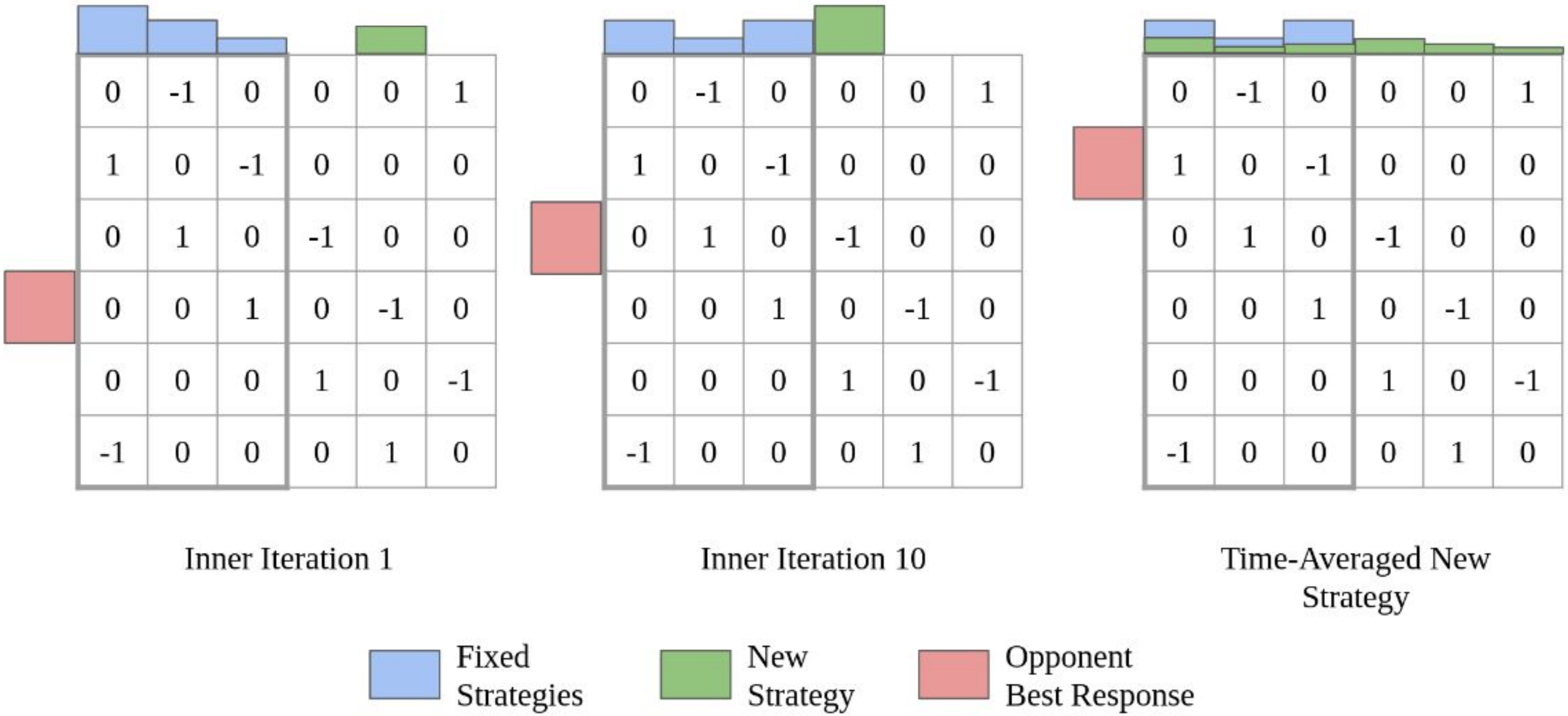
APSRO Results

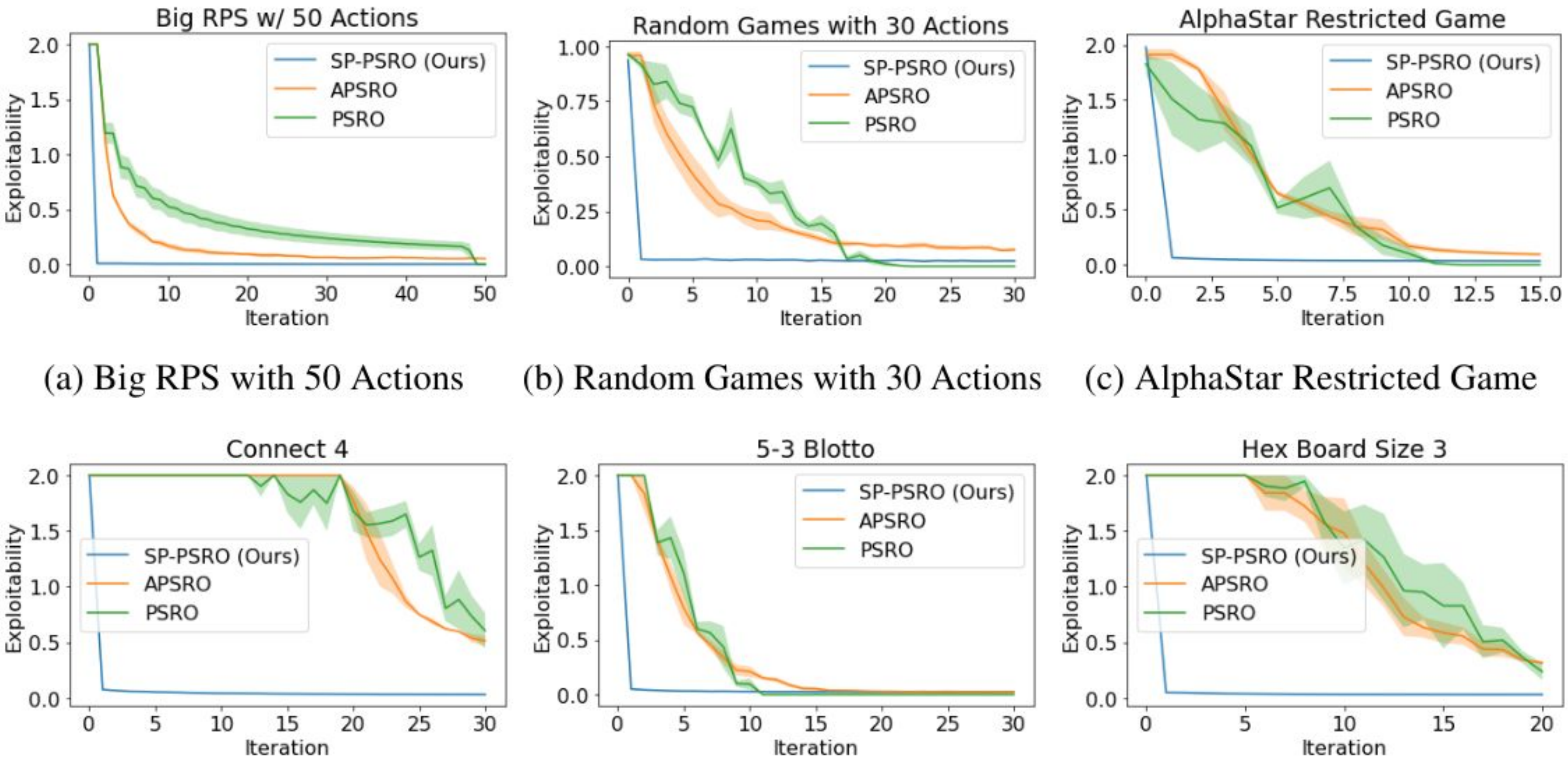


Which Strategies Should We Add?

- Best response to opponent restricted distribution not necessarily optimal
- Want to add strategy that minimizes exploitability of next iteration distribution
- Idea: include mixed strategies!
- New strategy trained in self-play against opponent best response

0	-1	0	0	0	1
1	0	-1	0	0	0
0	1	0	-1	0	0
0	0	1	0	-1	0
0	0	0	1	0	-1
-1	0	0	0	1	0

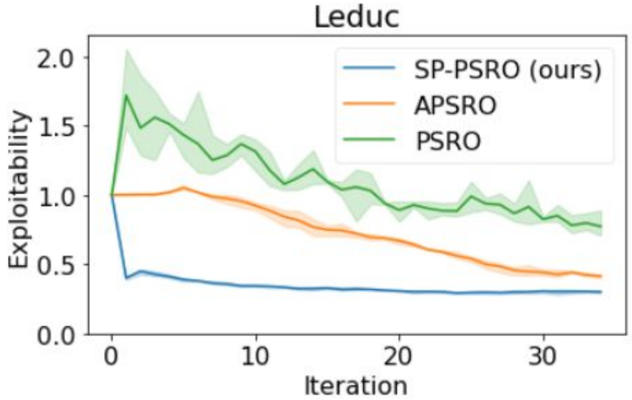

0 -1 0 0 0 1
1 0 -1 0 0 0
0 1 0 -1 0 0
0 0 1 0 -1 0
0 0 0 1 0 -1
-1 0 0 0 1 0
Inner Iteration 1
0 -1 0 0 0 1
1 0 -1 0 0 0
0 1 0 -1 0 0
0 0 1 0 -1 0
0 0 0 1 0 -1
-1 0 0 0 1 0
Inner Iteration 10
0 -1 0 0 0 1
1 0 -1 0 0 0
0 1 0 -1 0 0
0 0 1 0 -1 0
0 0 0 1 0 -1
-1 0 0 0 1 0
Time-Averaged New Strategy
Fixed Strategies
New Strategy
Opponent Best Response



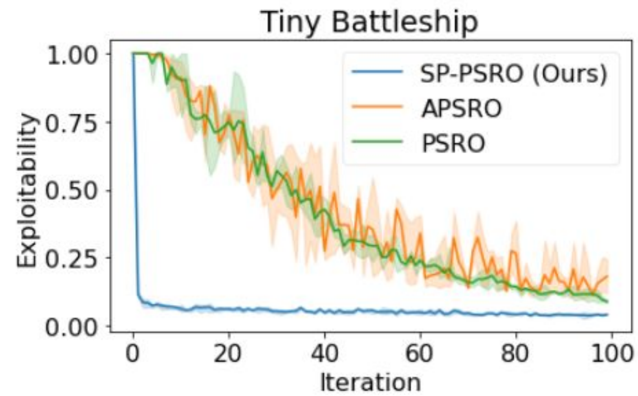
(a) Big RPS with 50 Actions (b) Random Games with 30 Actions (c) AlphaStar Restricted Game

(d) Connect 4 Restricted Game (e) 5-3 Blotto (f) Hex-3 Restricted Game

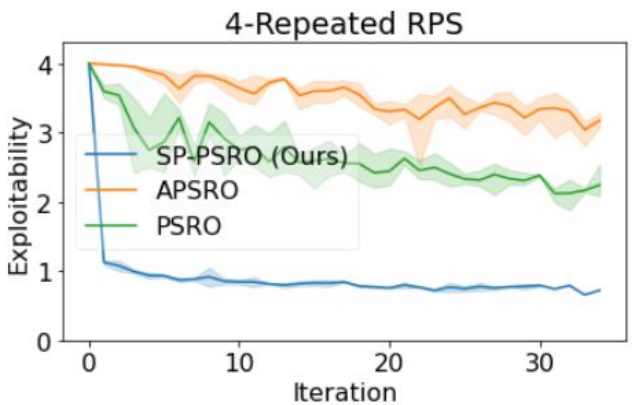
Figure 3: Normal-form games



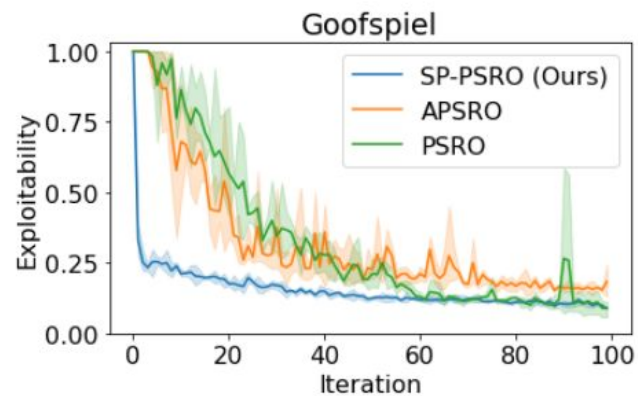
(a) Leduc Poker



(b) Battleship

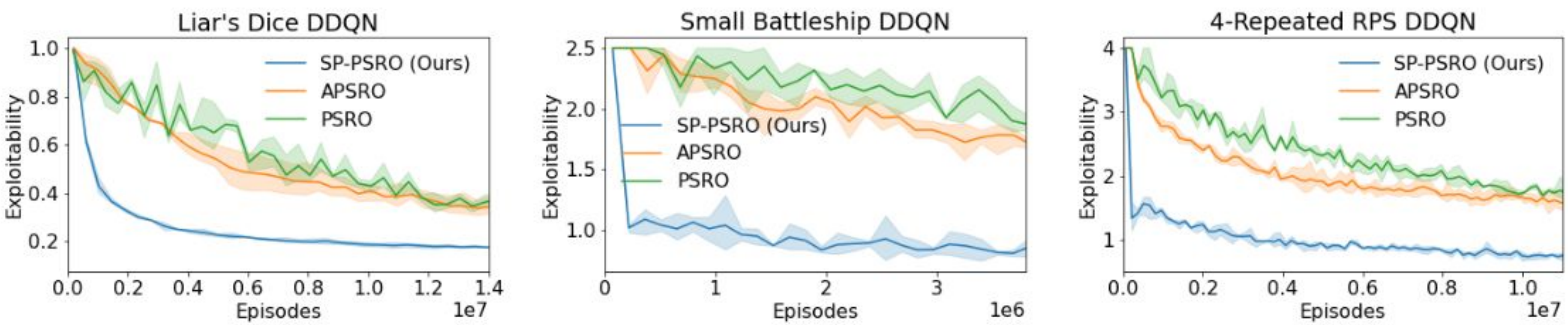


(c) Repeated RPS



(d) Goofspiel

Figure 4: Extensive-form games with tabular Q-learning best responses



(a) DRL Liars Dice (b) DRL Battleship (c) DRL Repeated RPS

Figure 5: Extensive-form games with DDQN best responses

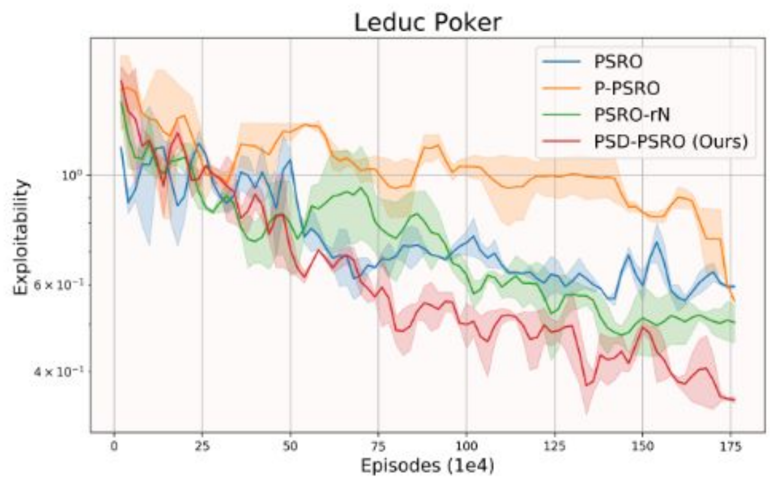
Diversity in PSRO

- Want to converge faster by adding “good” policies
- Good policies are ones that decrease exploitability
- One heuristic for this is diversity
- Can take the distance between two policies to be the KL

$$\pi_i^{t+1} = \arg\max_{\pi_i} \left\{ u(\pi_i, \sigma_{-i}^t) + \lambda \min_{\pi_i^k \in \mathcal{H}(\Pi_i^t)} \operatorname{dist}(\pi_i, \pi_i^k) \right\}$$

Yao, Liu, Fu, Yang, McAleer, Fu, Yang. Policy Space Diversity for Non-Transitive Games. NeurIPS ‘23

Results



	PSRO	$PSRO_{rN}$	P-PSRO	PSD-PSRO(OURS)
PSRO	-	0.613±0.019	0.469±0.034	0.422±0.025
$PSRO_{rN}$	0.387±0.019	-	0.412±0.030	0.358±0.019
P-PSRO	0.531±0.034	0.588±0.030	-	0.370±0.031
PSD-PSRO(OURS)	**0.578±0.025**	**0.642±0.019**	**0.630±0.031**	-

Table 1: The win rate of the row agents against the column agents on Goofspiel.

Yao, Liu, Fu, Yang, McAleer, Fu, Yang. Policy Space Diversity for Non-Transitive Games. NeurIPS '23