Deep Learning in Tree-Based Game Solving 3

Stephen McAleer

Single-agent reinforcement learning

Single-agent learning:

- Set of states S (information sets) with fixed "start state" $s_1 \in S$ (root infoset)
- Set of actions A
- We'll assume finite horizon: $S = S_1 \sqcup S_2 \sqcup \cdots S_H$, where $H =$ time horizon (depth of game tree), and $S_1 = \{s_1\}$
- Fixed environment (opponent/nature) given by transition functions $P : S_h \times A \to \Delta(S_{h+1})$ for each $h < H$. Playing action a in state s results in random next state s' w.p. $P(s'|s, a)$.
- Trajectory (history): $\tau = (s_1, a_1, s_2, a_2, \dots, a_{H-1}, s_H)$
- Policy (strategy): $\pi : S \to \Delta(A)$
- Reward (utility): $R : S_H \to \mathbb{R}$
  (assume for simplicity that reward is only received at the end)

Q-values, state values, and advantages

In extensive form, when multiplied by environment reach probability of infoset s, these are:

Define recursively:

$$Q^\pi(s,a) = \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} V^\pi(s')$$

"state-action value" — counterfactual value $u(a|s)$

$$V^\pi(s) = \begin{cases} R(s) & \text{if } s \in S_H \\ \mathop{\mathbb{E}}_{a \sim \pi(\cdot|s)} Q^\pi(s,a) & \text{otherwise} \end{cases}$$

"state value" — counterfactual value $u(s)$

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$$

"advantage" — immediate regret $g(a|s)$

Goal: find π maximizing expected reward

$$V^\pi(s_1) = \mathop{\mathbb{E}}_{\tau \sim \pi} R(s_H, a_H).$$

S, A small enough to iterate over
$\Rightarrow$ easy! (backwards induction)

S, A large $\Rightarrow$???

Policy gradient theorem

$$\nabla V^{\pi}(s_1) = \sum_{\tau} R(s_H)\nabla P(\tau|\pi)$$

$$= \sum_{\tau} R(s_H)\, P(\tau|\pi)\nabla \log P(\tau|\pi)$$

using $\nabla \log f(x) = \frac{\nabla f(x)}{f(x)}$

$$= \mathop{\mathbb{E}}_{\tau\sim\pi} R(s_H)\nabla \log P(\tau|\pi)$$

$$= \mathop{\mathbb{E}}_{\tau\sim\pi} \sum_{h=1}^{H-1} R(s_H)\nabla \log \pi(a_h|s_h)$$

$$= \mathop{\mathbb{E}}_{\tau\sim\pi} \sum_{h=1}^{H-1} A(s_h, a_h)\nabla \log \pi(a_h|s_h)$$

(won't show—same idea as "baselines")

Advantage actor-critic (A2C) (very roughly)

initialize policy π^1 to be uniform random

for $t = 1, \dots, T$:

of course, $\hat{V}(s_H) \coloneqq R(s_H)$

- train value function estimate $\tilde{V}^t \approx V^\pi$ using MSE:

$$\tilde{V}^t \coloneqq \arg\min_{\hat{V}} \mathop{\mathbb{E}}_{\tau \sim \pi^t} \sum_{h=1}^{H-1} \left[\hat{V}(s_h) - \hat{V}(s_{h+1})\right]^2$$

- train policy π^{t+1} by taking gradient steps according to the policy gradient theorem:

$$\nabla V^\pi(s_1) = \mathop{\mathbb{E}}_{\tau \sim \pi^t} \sum_{h=1}^{H-1} \hat{A}(s_h, a_h) \nabla \log \pi(a_h | s_h)$$

Problem: Variance in gradients can be very large, so π can change very fast $\Rightarrow$ training can be unstable

where

$$\tilde{A}^t(s,a) \coloneqq \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \tilde{V}^t(s') - \tilde{V}^t(s)$$

is an advantage function estimate

Proximal policy optimization (PPO) (very roughly)

initialize policy π^1 to be uniform random

for $t = 1, \dots, T$:

of course, $\hat{V}(s_H) \coloneqq R(s_H)$

- train value function estimate $\tilde{V}^t \approx V^\pi$ using MSE:

$$\tilde{V}^t \coloneqq \arg\min_{\hat{V}} \mathbb{E}_{\tau \sim \pi^t} \sum_{h=1}^{H-1} \left[\hat{V}(s_h) - \hat{V}(s_{h+1})\right]^2$$

- train policy π^{t+1} according to:

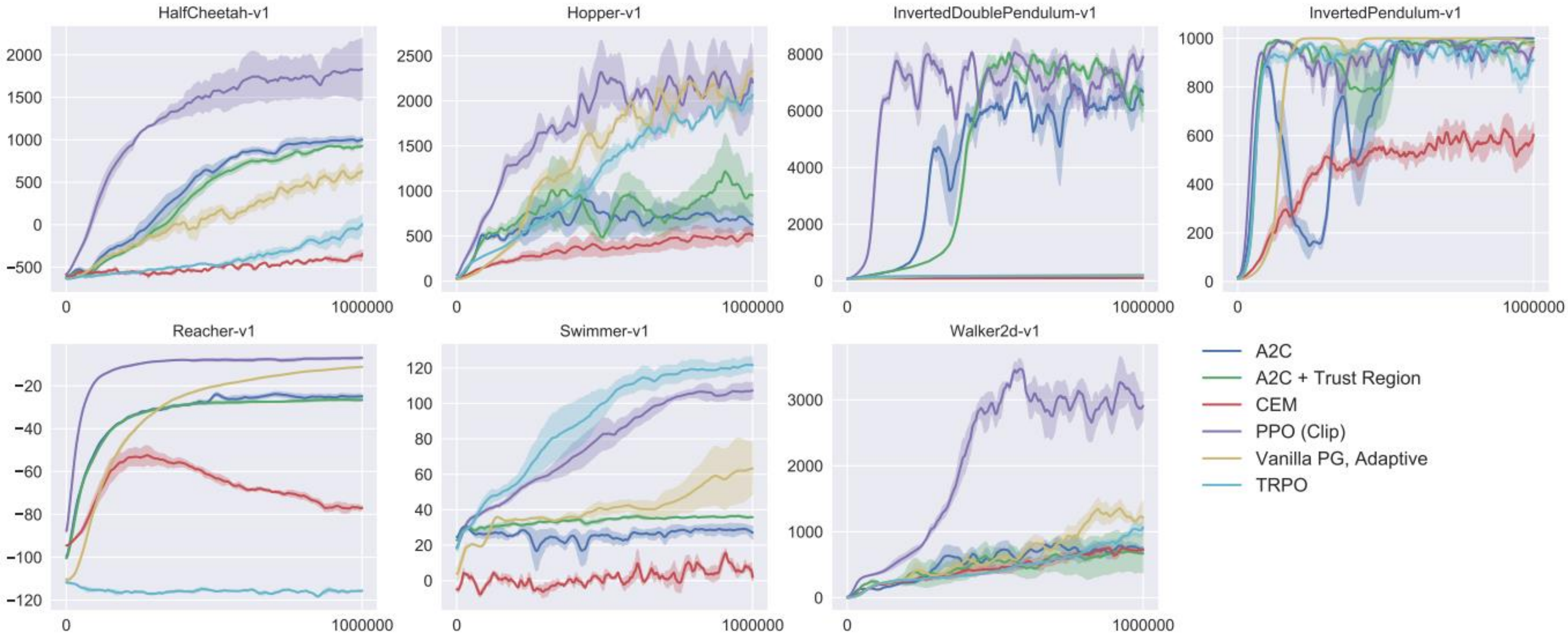
$$\pi^{t+1} \coloneqq \arg\max_{\hat{\pi}} \mathbb{E}_{\tau \sim \pi^t} \sum_{h=1}^{H-1} \begin{cases} \min\left\{\frac{\hat{\pi}(a_h|s_h)}{\pi^t(a_h|s_h)}, 1+\epsilon\right\} \hat{A}^t(s_h, a_h) & \text{if } \hat{A}^t(s_h, a_h) > 0 \\ \max\left\{\frac{\hat{\pi}(a_h|s_h)}{\pi^t(a_h|s_h)}, 1-\epsilon\right\} \hat{A}^t(s_h, a_h) & \text{if } \hat{A}^t(s_h, a_h) < 0 \end{cases}$$

where

$$\tilde{A}^t(s, a) \coloneqq \mathbb{E}_{s' \sim P(\cdot|s,a)} \tilde{V}^t(s') - \tilde{V}^t(s)$$

is an advantage function estimate

PPO is great*



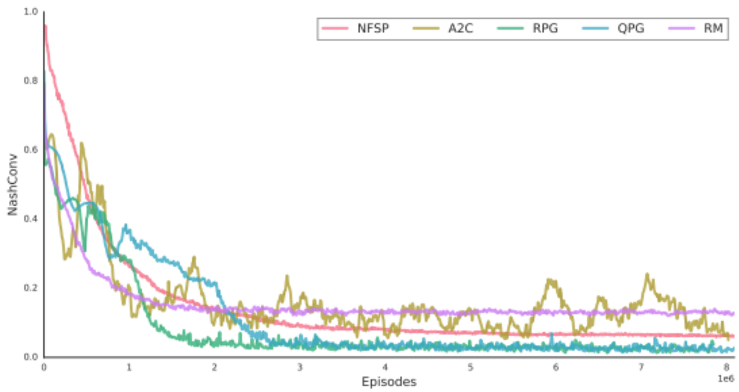
Best large-scale single-agent RL algorithm right now!

***Very very sensitive** to hyperparameters… hard to use in practice…

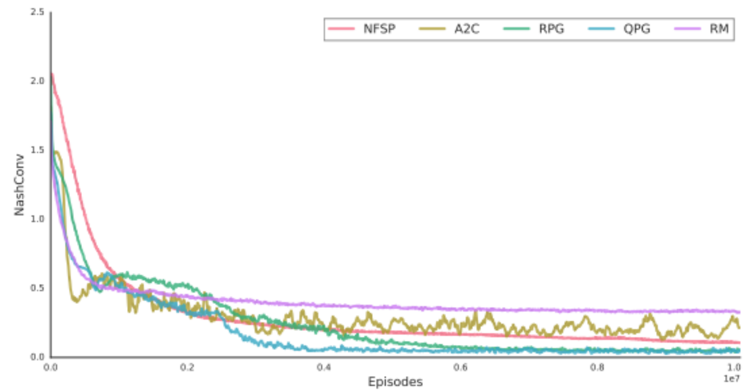
Do these algorithms work for games?

Certainly not in theory. In practice… kind of, at small scale?

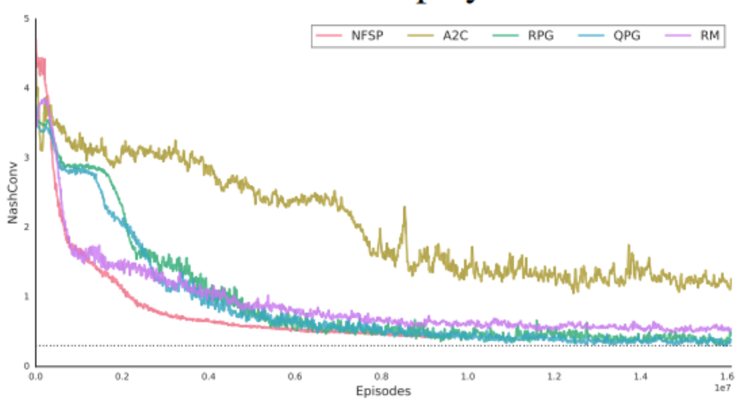
(but probably at this scale you should just use PCFR+ instead…)



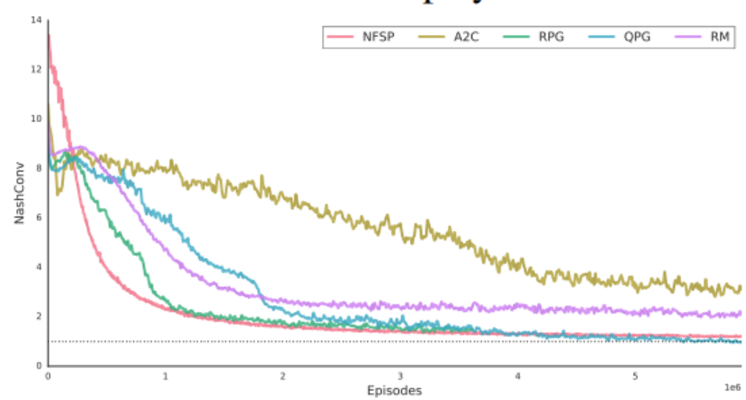
NashConv in 2-player Kuhn



NashConv in 3-player Kuhn



NashConv in 2-player Leduc



NashConv in 3-player Leduc

References

MCCFR: Marc Lanctot, Kevin Waugh, Martin Zinkevich, Michael Bowling (NeurIPS 2009) "Monte Carlo sampling for regret minimization in extensive games"

Simplified martingale-based presentation and improved bound in this lecture due to Gabriele Farina, Christian Kroer, Tuomas Sandholm (ICML 2020) "Stochastic regret minimization in extensive-form games"

Martin Schmid, Neil Burch, Marc Lanctot, Matej Moravcik, Rudolf Kadlec, Michael Bowling (AAAI 2019) "Variance Reduction in Monte Carlo Counterfactual Regret Minimization (VR-MCCFR) for Extensive Form Games using Baselines"

Noam Brown, Adam Lerer, Sam Gross, Tuomas Sandholm (ICML 2019) "Deep Counterfactual Regret Minimization"

Deep CFR with variance reduction: Eric Steinberger, Adam Lerer, Noam Brown (arXiv 2020) "DREAM: Deep regret minimization with advantage baselines and model-free learning"

Stephen McAleer, Gabriele Farina, Marc Lanctot, Tuomas Sandholm (ICLR 2023) "Eschewing Importance Sampling in Games by Computing a History Value Function to Estimate Regret"

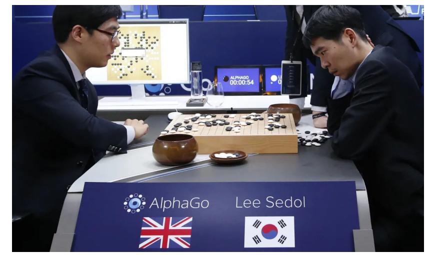
John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov (arXiv 2017) "Proximal Policy Optimization Algorithms"

Sriram Srinivasan, Marc Lanctot, Vinicius Zambaldi, Julien Perolat, Karl Tuyls, Remi Munos, Michael Bowling (NeurIPS 2018) "Actor-Critic Policy Optimization in Partially Observable Multiagent Environments"

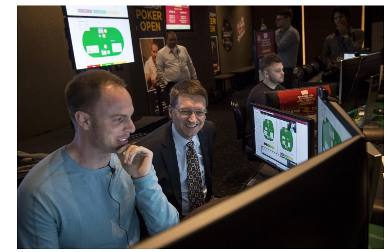
Games in AI

Backgammon
1992

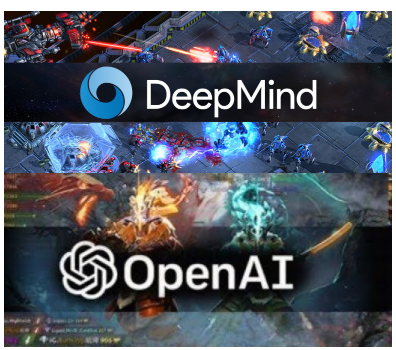
Chess
1997



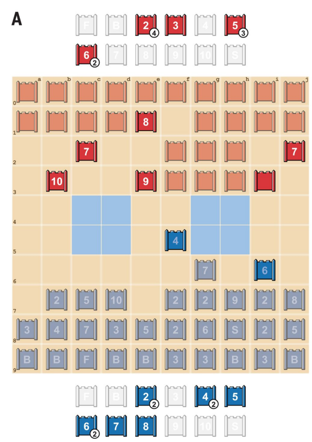
Go
2016



Poker
2017/2019



Starcraft/Dota
2019

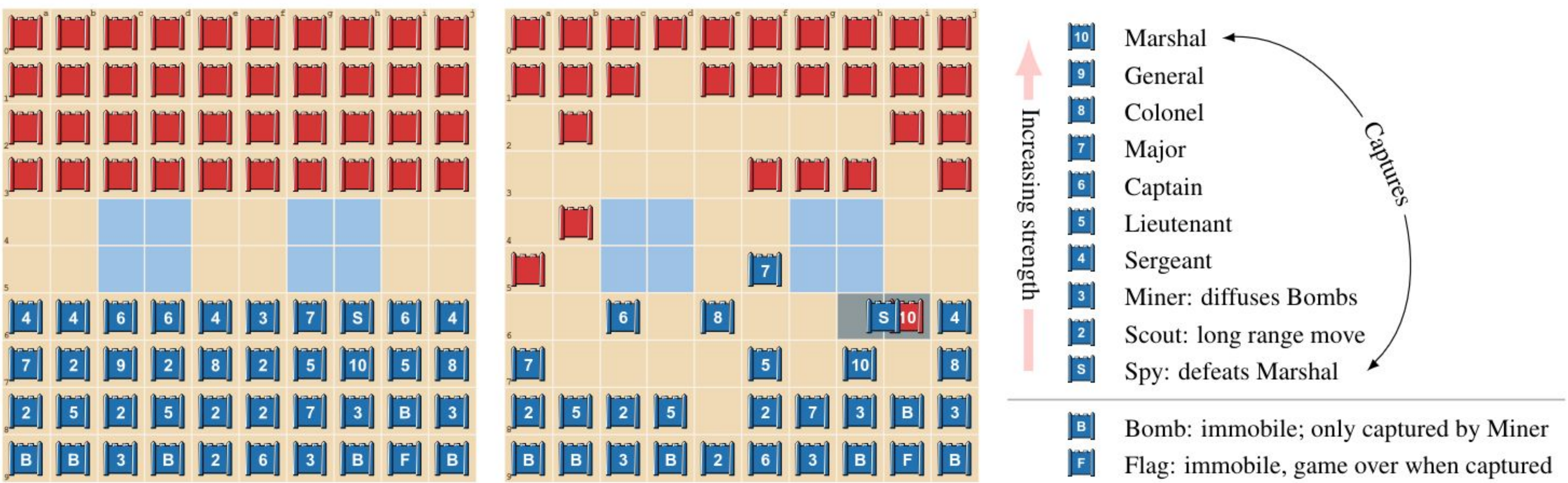


Stratego
2022

Diplomacy
2022

Stratego

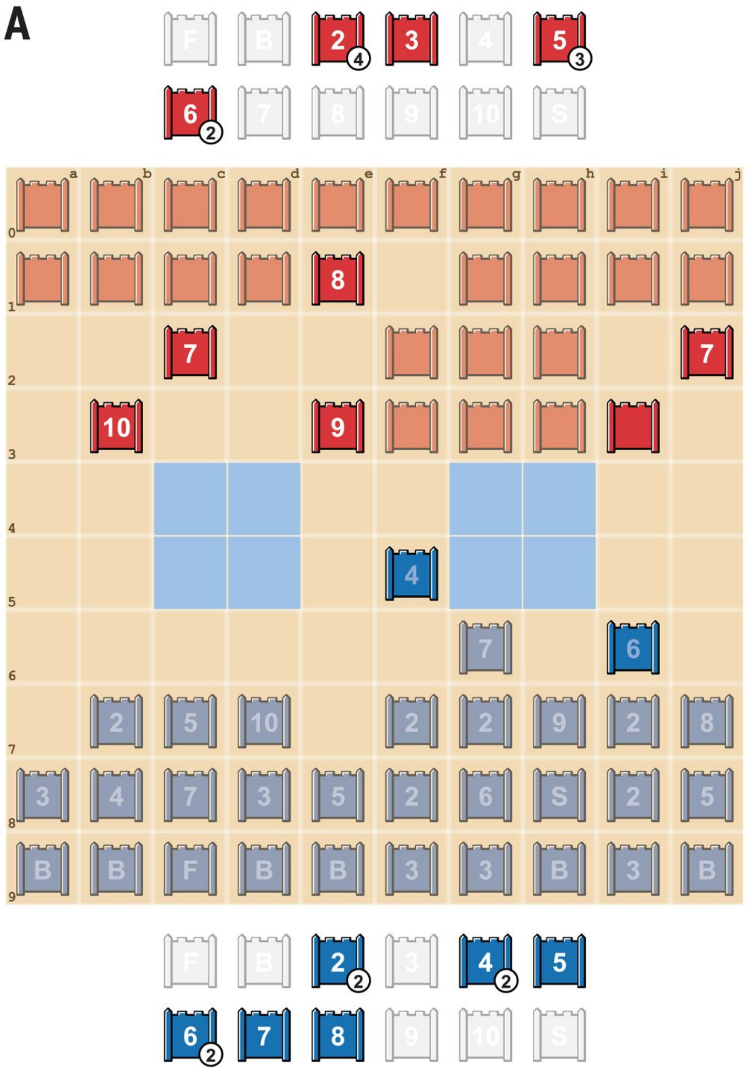
- Pieces are numbered from 2 to 10 (Also a spy, bomb and **flag)**
- Higher numbers capture lower numbers (Exceptions: spys, bombs)
- First, both players place their pieces (**Can't see opponents pieces)**
- Each piece moves one square (Exception: 2)
- If your piece is captured, you see the other piece number
- Objective is to capture the opponent's flag



Phase 1: Private deployment **Phase 2:** Game play Piece types

Stratego

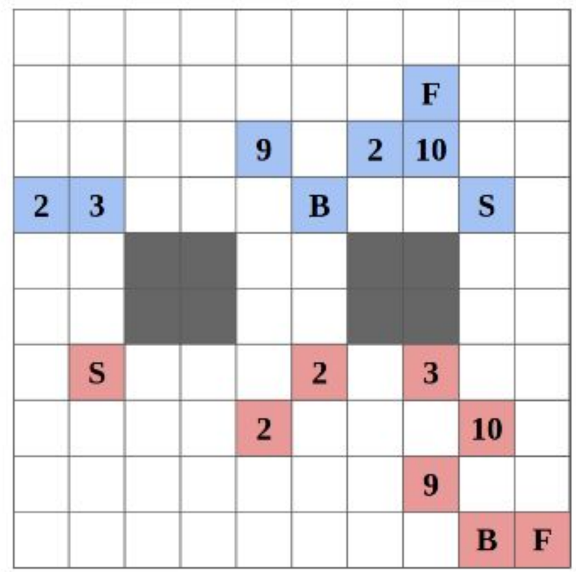
- Two challenges: **size** and **imperfect information**
- Size: order of 10^{535} nodes
  - Texas hold 'em: 10^{164} nodes
  - Go: 10^{360} nodes
- Imperfect information
  - 10^{66} possible deployments
  - Can't use perfect-info search
  - Bluffing, mixing are important
  - Gathering and hiding information very important
- Compared to video games, decisions are made deliberately
  - Doesn't just test reaction time and instincts



Stratego

- Existing approaches have hand-coded rules and play at an amateur level
- PSRO-based approach got SOTA on Barrage Stratego in 2020
    - Still played at an amateur level

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



McAleer*, Lanier*, Fox, Baldi. Pipeline PSRO: A Scalable Approach for Finding Approximate Nash Equilibria in Large Games. NeurIPS 2020

Finding Equilibrium via Regularization

- Continuous-time Follow-the-Regularized Leader (FoReL)

$$y_t^i(a^i) = \int_0^t Q_{\pi_s}^i(a^i)ds \quad \text{and} \quad \pi_t^i = \underset{p \in \Delta A}{\arg\max}\, \Lambda^i(p, y_t^i)$$

$$\Lambda^i(p, y) = \langle y, p \rangle - \phi_i(p)$$

$$\phi_i^*(y) = \max_p \Lambda^i(p, y)$$

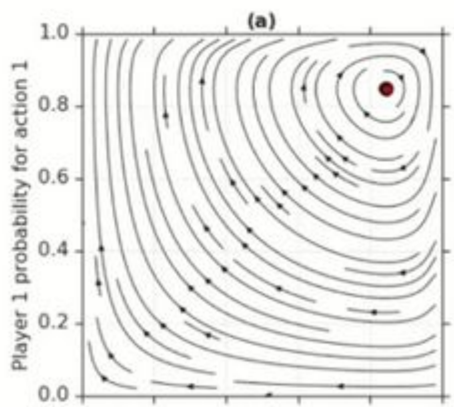
- Motivation: want to get last-iterate convergence

Perolat et al. From Poincare Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization. ICML 2021.

Finding Equilibrium via Regularization

- In two-player zero-sum games, the Nash Gap (exploitability) is preserved, so FoReL is recurrent

$$J(y) = \sum_{i=1}^{2} \left[\phi_i^*(y_i) - \langle y_i, \pi_i^* \rangle \right]$$



Perolat et al. From Poincare Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization. ICML 2021.

Finding Equilibrium via Regularization

- If we modify the game to have this new policy-dependent reward function

$$r^i_\pi(a) = r^i(a^i, a^{-i}) - \eta \log \frac{\pi^i(a^i)}{\mu^i(a^i)} + \eta \log \frac{\pi^{-i}(a^{-i})}{\mu^{-i}(a^{-i})}$$

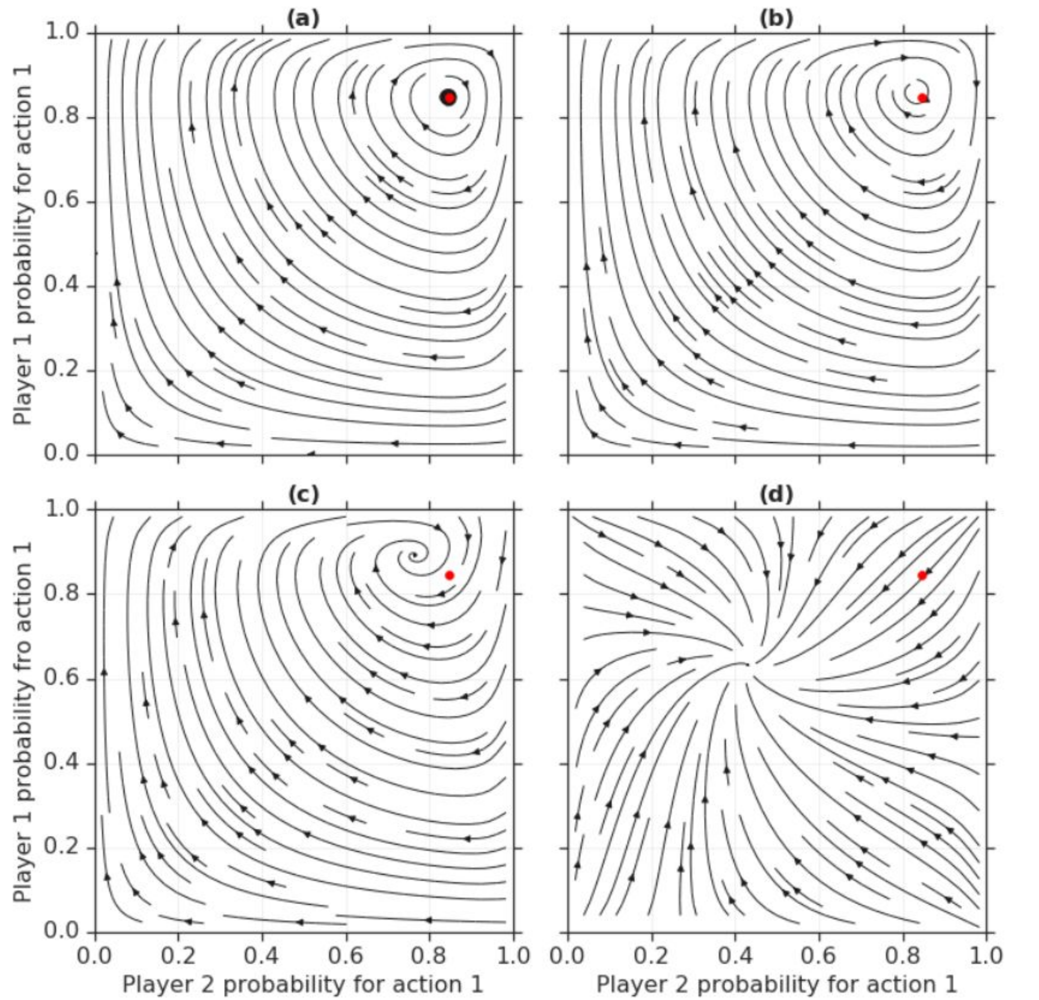
- Then FoReL is convergent

$$\frac{d}{dt} J(y) = \sum_{i=1}^{2} \underbrace{[V^i_{\pi^i_t, \pi^{*-i}} - V^i_{\pi^*}]}_{\leq 0 \text{ because } \pi^* \text{ is a Nash}} - \eta \sum_{i=1}^{2} KL(\pi^{*i}, \pi^i_t)$$

Perolat et al. From Poincare Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization. ICML 2021.

Finding Equilibrium via Regularization

- However, FoReL converges to a biased solution
- Plot shows eta= 0, 0.5, 1, and 10



Perolat et al. From Poincare Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization. ICML 2021.

Finding Equilibrium via Regularization

- Solve the original game by iteratively using last policy as the reference policy

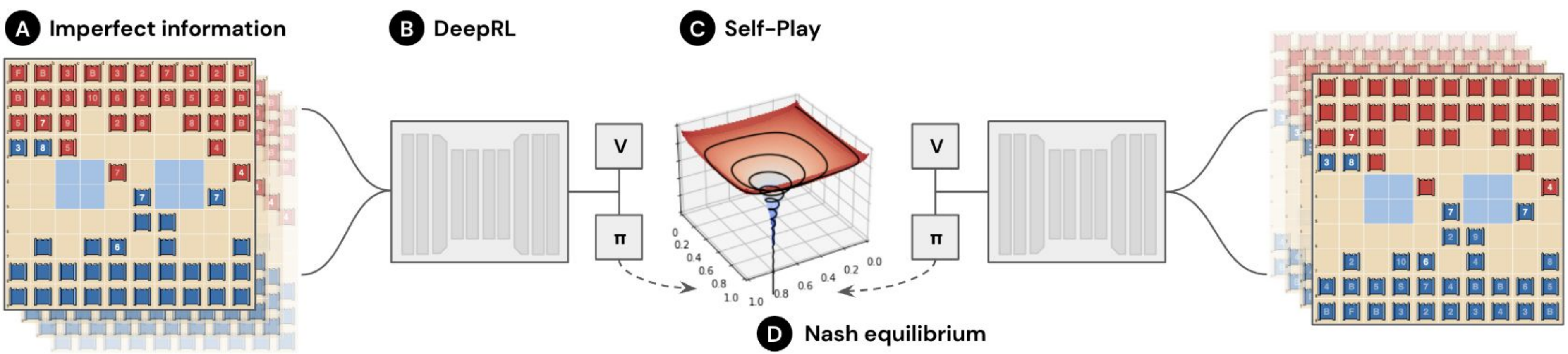
$$r^i_{k,\pi}(h,a) = r^i(a^i, a^{-i}) - \eta \log \frac{\pi^i(a^i)}{\pi^i_{k-1}(a^i)} + \eta \log \frac{\pi^{-i}(a^{-i})}{\pi^{-i}_{k-1}(a^{-i})}$$

- This procedure monotonically gets closer to Nash

Perolat et al. From Poincare Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization. ICML 2021.

DeepNash

- Two components
  - NeuRD
  - Regularized Nash Dynamics (R-NaD)



Replicator dynamics: $\frac{d}{d\tau}\pi^i_\tau(a^i) = \pi^i_\tau(a^i)\left[Q^i_{\pi_\tau}(a^i) - \sum_{b^i}\pi^i_\tau(b^i)Q^i_{\pi_\tau}(b^i)\right]$

Reward transformation: $r^i(\pi^i, \pi^{-i}, a^i, a^{-i}) = r^i(a^i, a^{-i}) - \eta \log\left(\frac{\pi^i(a^i)}{\pi^i_{\text{reg}}(a^i)}\right) + \eta \log\left(\frac{\pi^{-i}(a^{-i})}{\pi^{-i}_{\text{reg}}(a^{-i})}\right)$

Perolat et al. Mastering the Game of Stratego with Model-Free Multiagent Reinforcement Learning. Science 2022

DeepNash

- Regularized Nash Dynamics (R-NaD)
    - Same as in previous paper

		Player 2	
		Head: H	Tail: T
Player 1	Head: H	1	-1
	Tail: T	-1	1

(a) Matching pennies

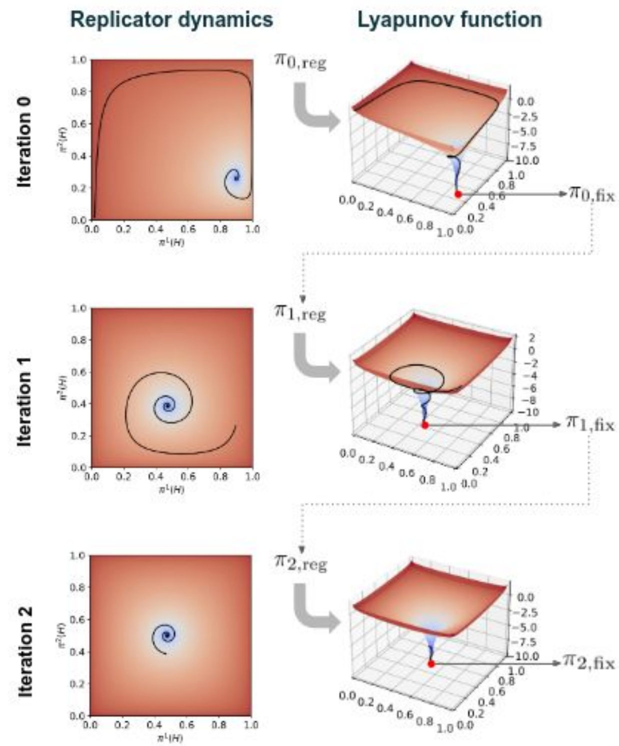
R-NaD Iteration

Start with an arbitrary regularization policy: $\pi_{0,\text{reg}}$

1. Reward transformation: Construct the transformed game with: $\pi_{n,\text{reg}}$
2. Dynamics: Run the replicator dynamics until convergence to: $\pi_{n,\text{fix}}$
3. Update: Set the regularization policy:
$$\pi_{n+1,\text{reg}} = \pi_{n,\text{fix}}$$

Repeat steps until convergence

(b) Algorithmic steps



(c) Dynamics and Lyapunov function

Figure 2: The R-NaD learning algorithm illustrated with the matching pennies game

DeepNash

- Same reward transformation as before

$$r^i(\pi^i, \pi^{-i}, a^i, a^{-i}) \;=\; r^i(a^i, a^{-i}) - \eta \log(\frac{\pi^i(a^i)}{\pi^i_{\text{reg}}(a^i)}) + \; \eta \log(\frac{\pi^{-i}(a^{-i})}{\pi^{-i}_{\text{reg}}(a^{-i})})$$

- Learn value function via V-Trace
- Learn policy via NeuRD

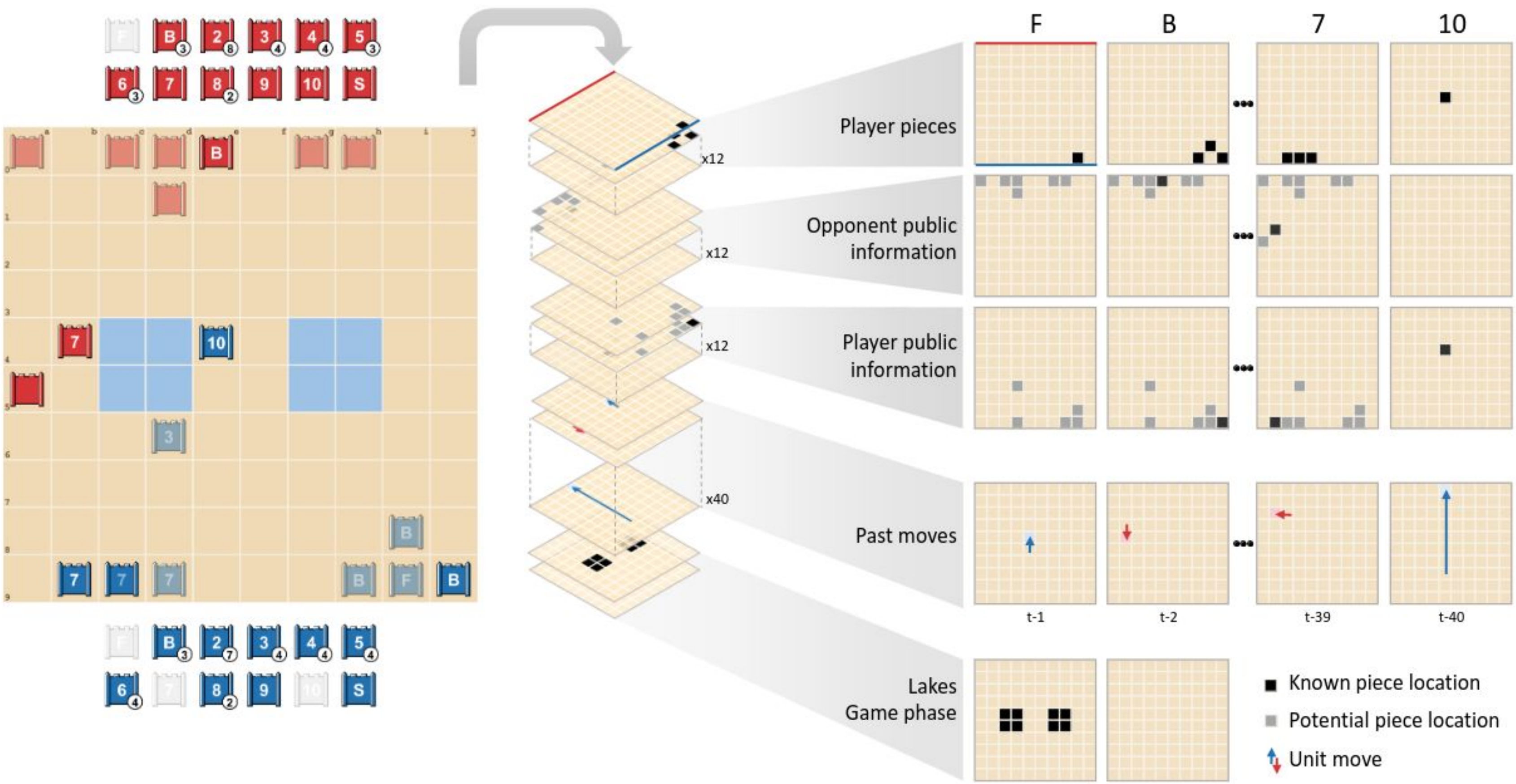
$$\Lambda_n = -\left[\mathrm{lr}_n \nabla l_{\text{critic}}(\theta_n) + \sum_{i=1}^{2} \frac{1}{t_{\text{effective}}} \sum_{t=0}^{t_{\text{effective}}} \sum_{a} \hat{\nabla}\theta(l_{\theta_n}(a, o_t)\mathrm{Clip}\left(Q^{\psi_t}_{t,n}(a, o_t), c_{\text{clip NeuRD}}\right), \mathrm{lr}_n, \beta)\right]$$

- Adapts IMPALA to parallelize

Perolat et al. Mastering the Game of Stratego with Model-Free Multiagent Reinforcement Learning. Science 2022

DeepNash

- Neural network input doesn't include full observation history, but a lot of it



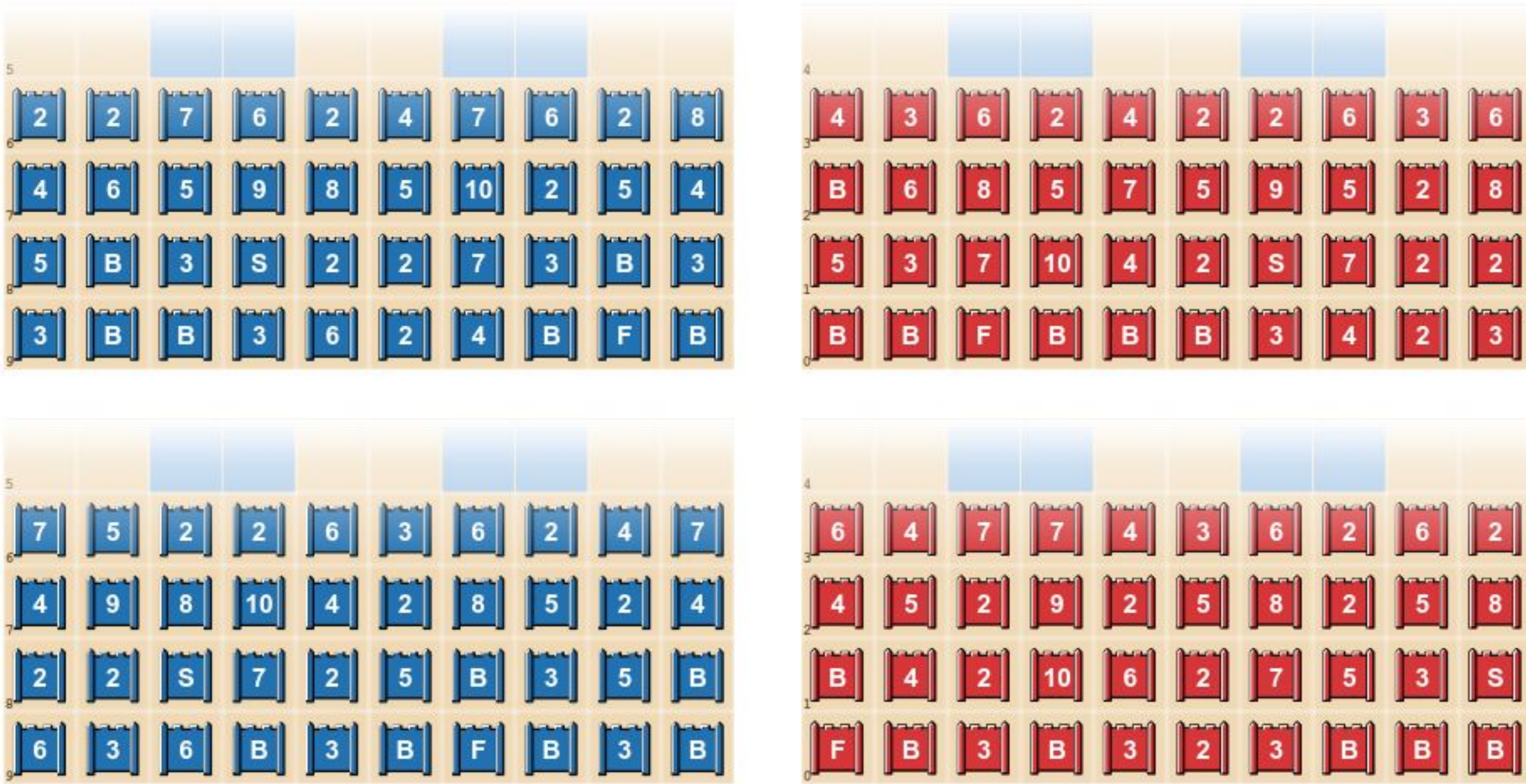
Results

Opponent	Number of Games	Wins	Draws	Losses
Probe	30	100.0%	0.0%	0.0%
Master of the Flag	30	100.0%	0.0%	0.0%
Demon of Ignorance	800	97.1%	1.8%	1.1%
Asmodeus	800	99.7%	0.0%	0.3%
Celsius	800	98.2%	0.0%	1.8%
Celsius1.1	800	97.9%	0.0%	2.1%
PeternLewis	800	99.9%	0.0%	0.1%
Vixen	800	100.0%	0.0%	0.0%

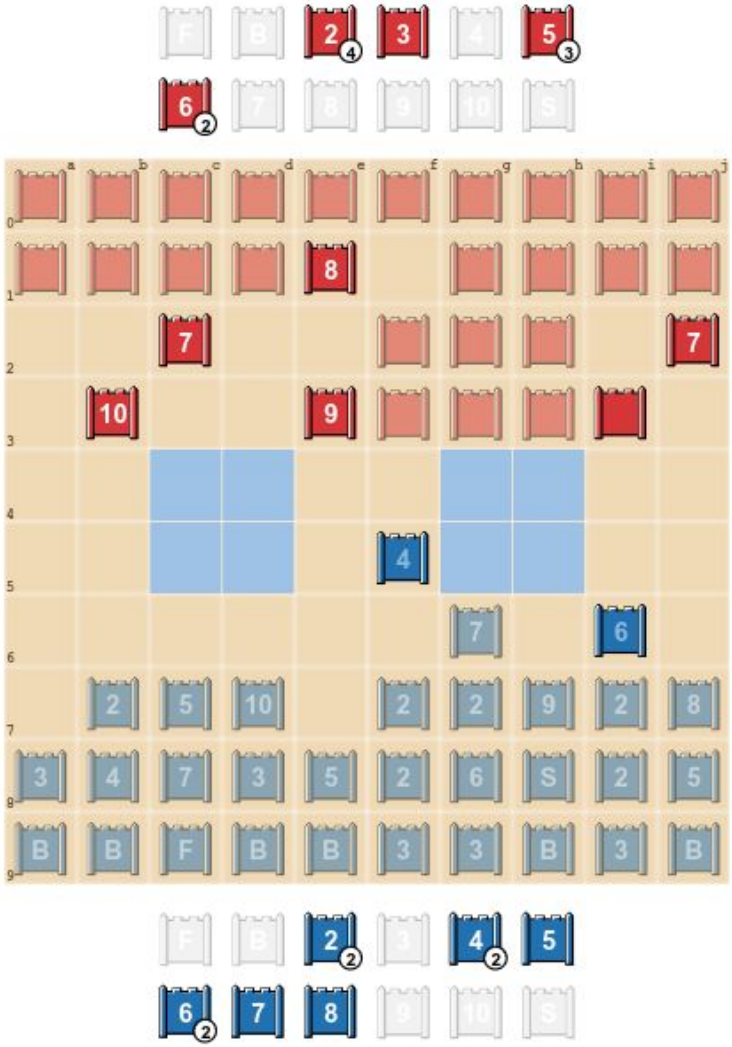
Expert-Level Performance: Won 84% of games on online server, placing it 3rd all-time.

Perolat et al. Mastering the Game of Stratego with Model-Free Multiagent Reinforcement Learning. Science 2022

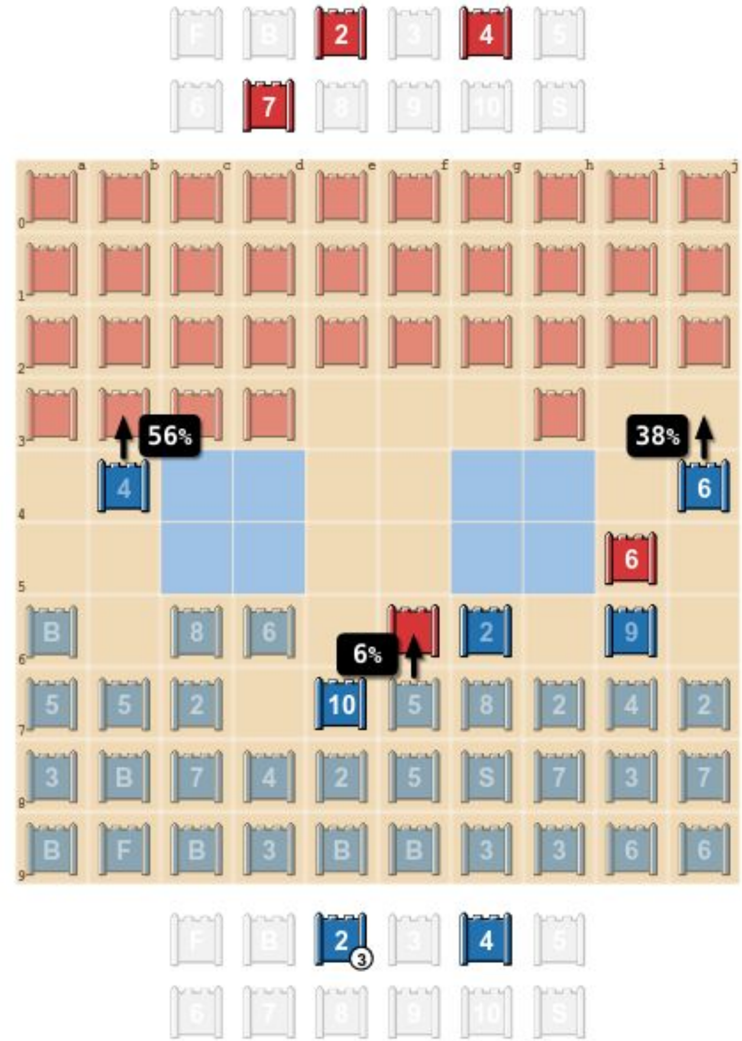
Results



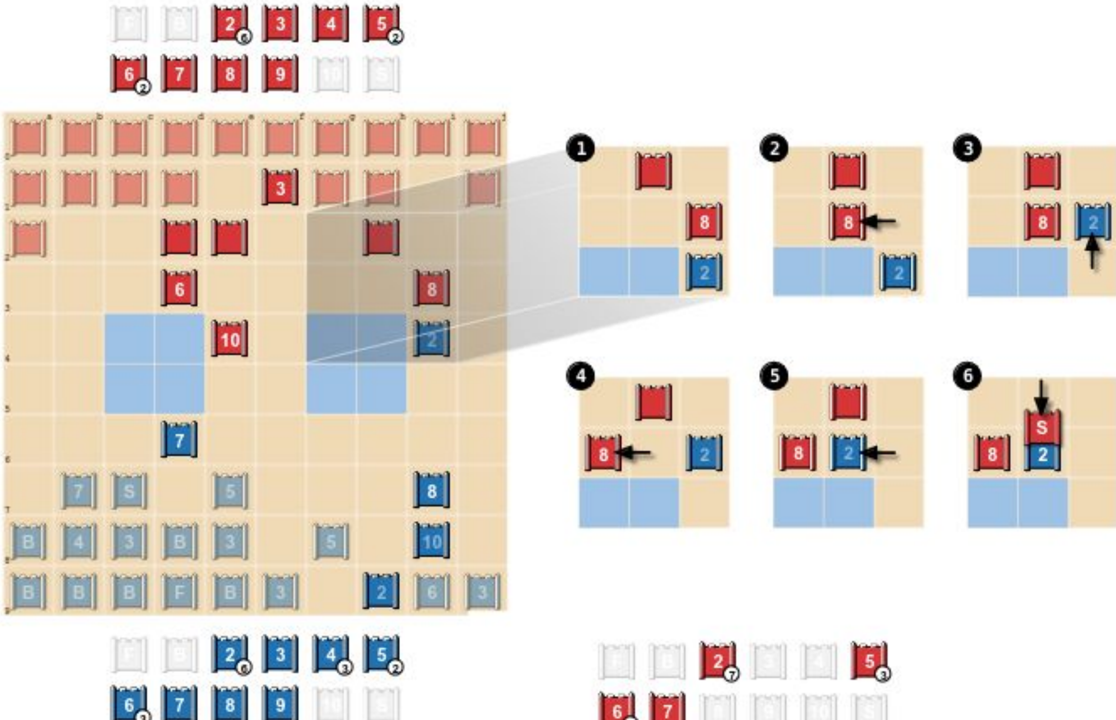
(a) Four example deployments *DeepNash* played on Gravon.



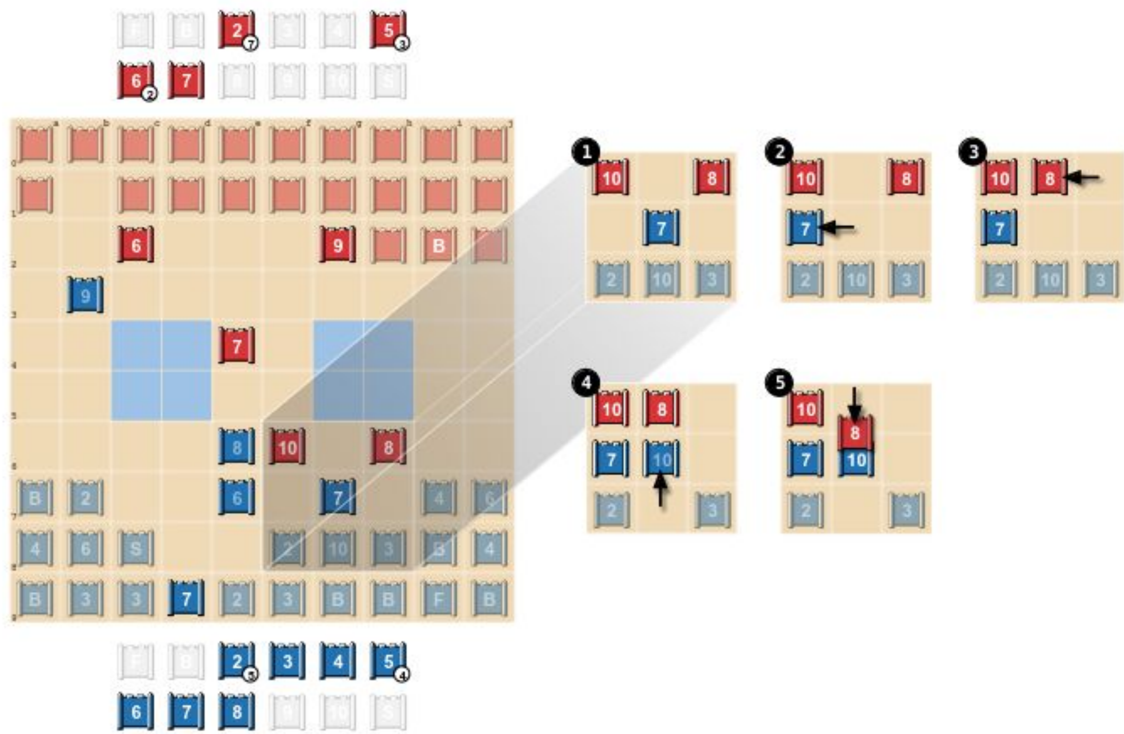
(b) While Blue is behind a 7 and 8, none of its pieces are revealed and only two pieces moved. As a result *DeepNash* assesses its chance of winning to be still around 70% (Blue indeed won this match).



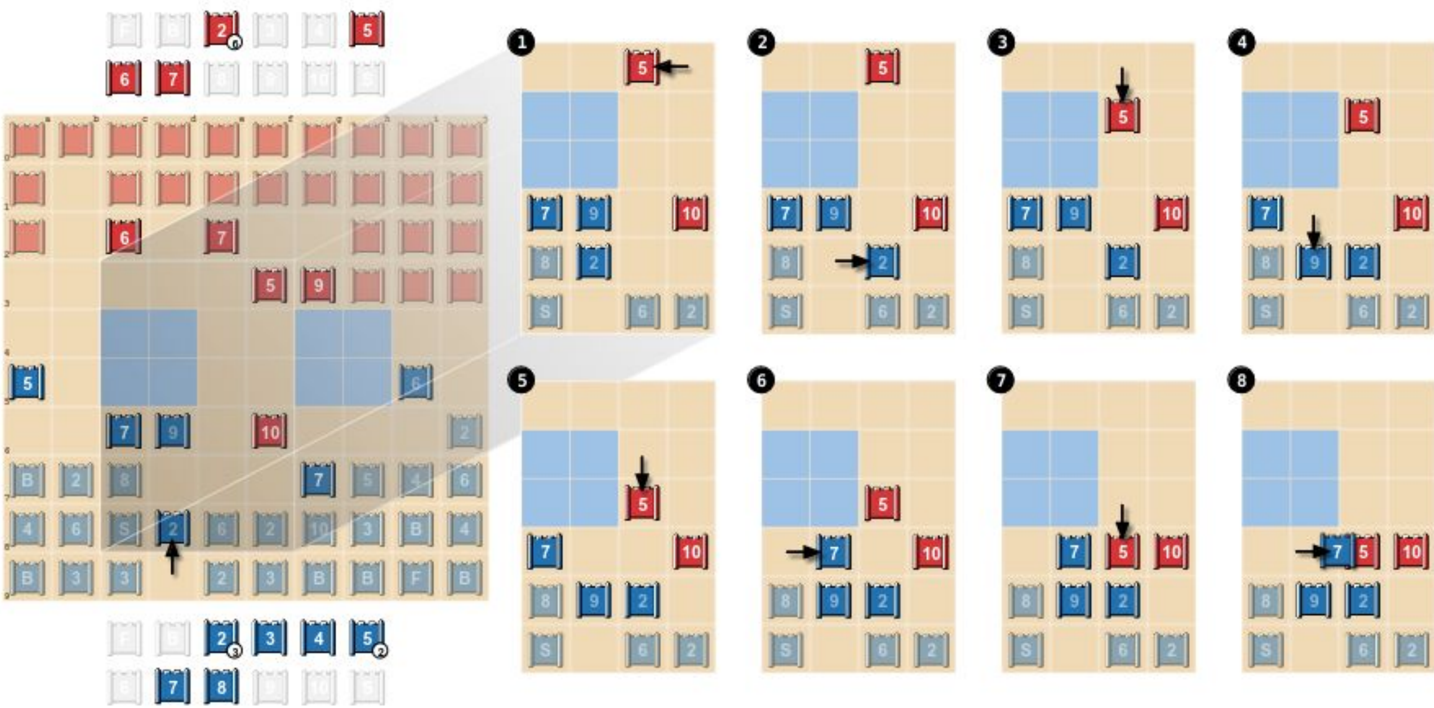
(c) Blue to move. *DeepNash*'s policy supports three moves at this state, with the indicated probabilities (the move on the right was played in the actual match). While Blue has the opportunity to capture the opponent's 6 with its 9, this move is not considered by *DeepNash*, likely because the protection of 9's identity is assessed to be more important than the material gain.



(a) Positive bluffing.



(b) Negative bluffing.



(c) *DeepNash* makes a Scout (2) behave like a Spy and gains material.

Figure 5: Illustration of *DeepNash* bluffing.

Perolat et al. Mastering the Game of Stratego with Model-Free Multiagent Reinforcement Learning. Science 2022

What is mirror descent?

- Generalization of gradient descent to different notions of distance

$$x_{t+1} = \arg\min_x \langle g, x \rangle + \frac{1}{\eta} \mathrm{B}(x, x_t)$$

- Negative Entropy (policy space):

$$\pi_{t+1} = \arg\max_\pi \langle q, \pi \rangle - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t)$$

What is **magnetic** mirror descent?

- Generalization of **regularized** gradient descent to different notions of distance

$$x_{t+1} = \arg\min_x \langle g, x \rangle + \frac{1}{\eta} \mathrm{B}(x, x_t) + \alpha \mathrm{B}(x, z)$$

- Negative Entropy (policy space):

$$\pi_{t+1} = \arg\max_\pi \langle q, \pi \rangle - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t) - \alpha \mathrm{KL}(\pi, \rho)$$

$$\propto \left[\pi_t e^{\eta q} \rho^{\alpha\eta}\right]^{\frac{1}{1+\alpha\eta}}$$

Theoretical Grounding

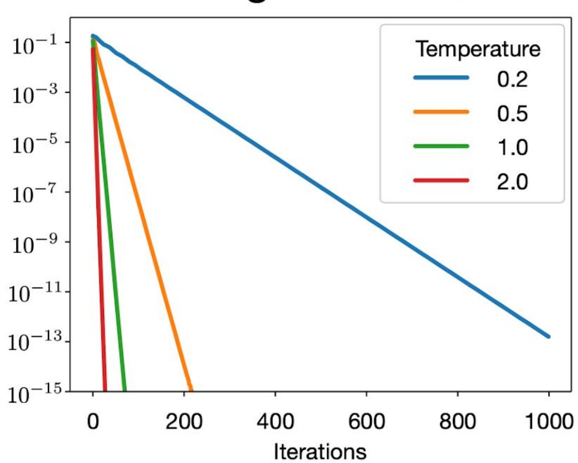
In two-player zero-sum one-shot games, if

$$\eta \leq \alpha / L^2$$

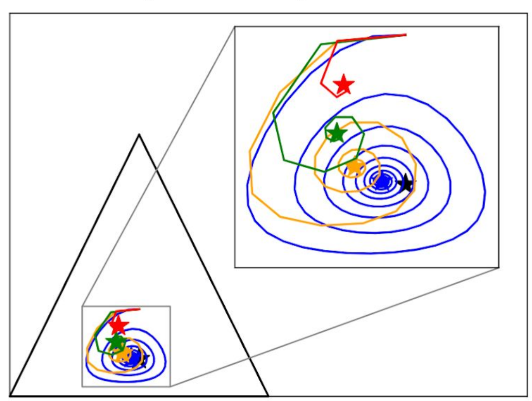
magnetic mirror descent converges exponentially fast to a

regularized equilibrium in self play

KL Divergence to QRE



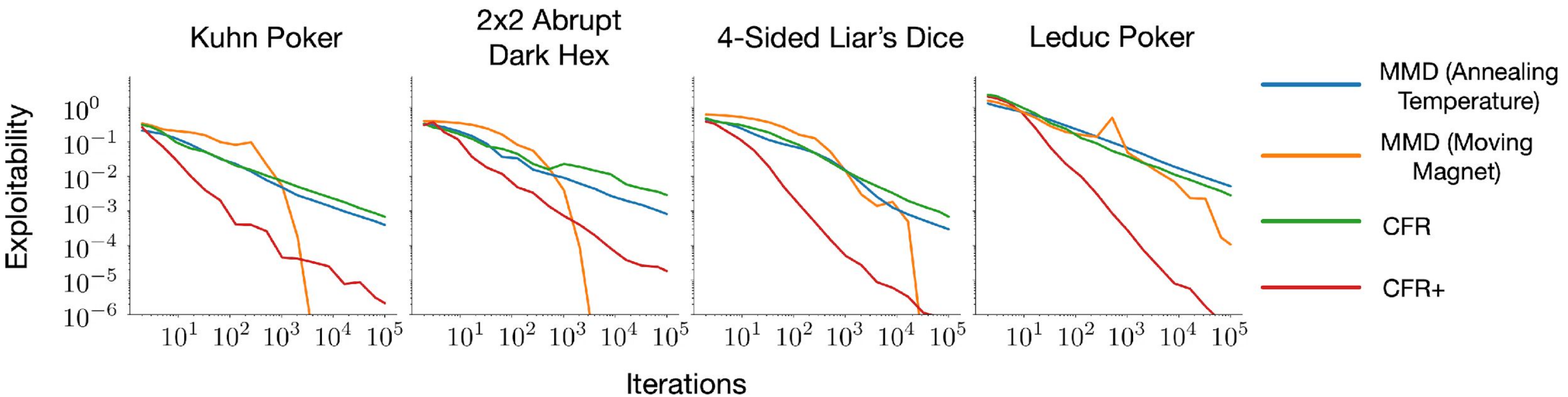
Simplex Trajectories



Payoff Matrix

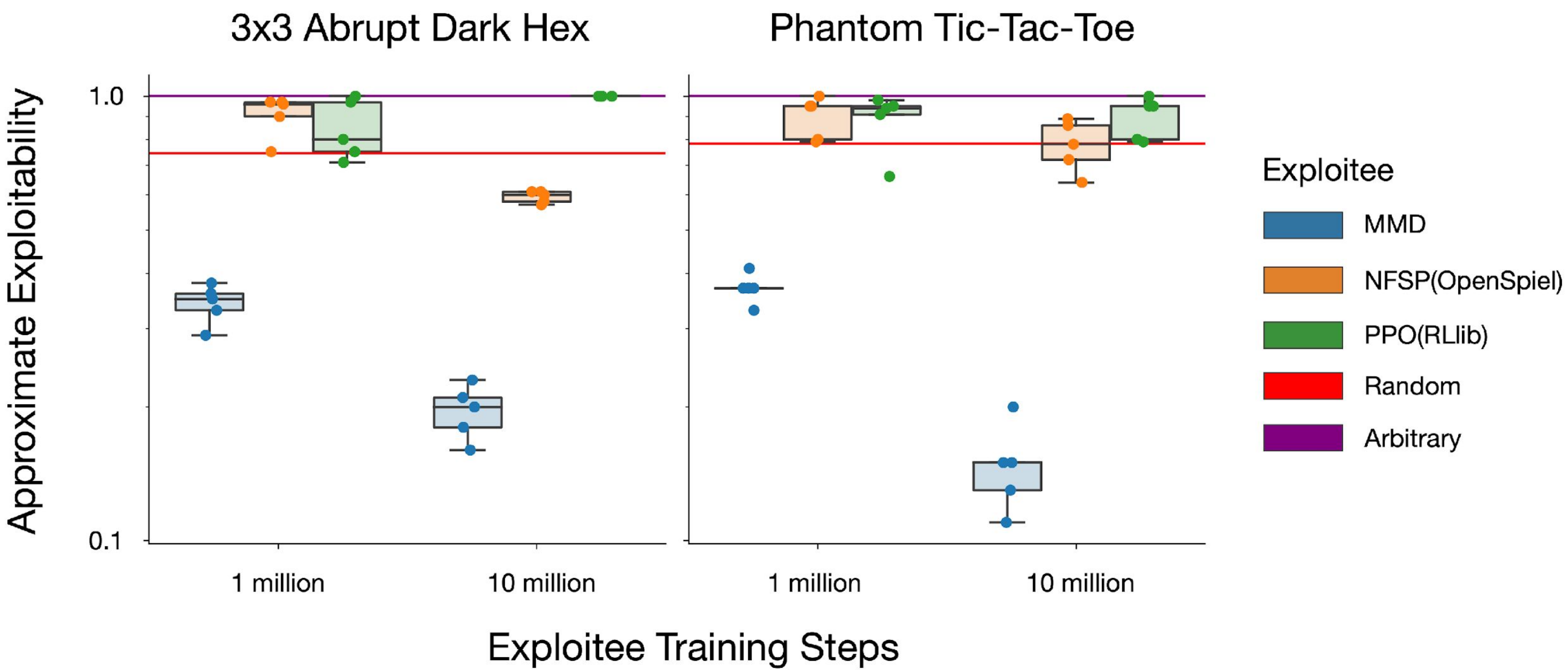
	R	P	S
R	0	-1	3
P	1	0	-3
S	-3	3	0

Comparison Against CFR



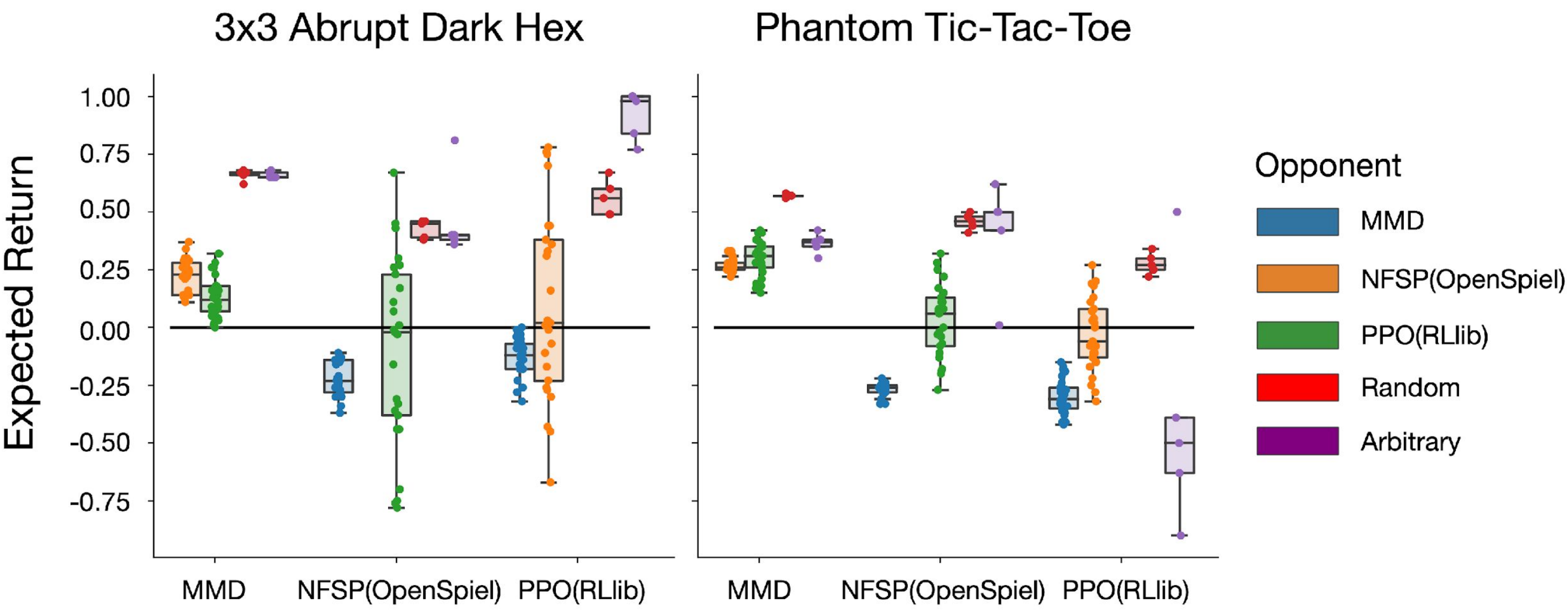
Sokota et al. A Unified Approach to Reinforcement Learning, Quantal Response Equilibria, and Two-Player Zero-Sum Games. ICLR 2023

Deep RL Experiments: Approximate Exploitability



Sokota et al. A Unified Approach to Reinforcement Learning, Quantal Response Equilibria, and Two-Player Zero-Sum Games. ICLR 2023

Deep RL Experiments: Head-to-Head Matchups



Sokota et al. A Unified Approach to Reinforcement Learning, Quantal Response Equilibria, and Two-Player Zero-Sum Games. ICLR 2023