DeepMind

Game-Theoretic Approaches for Multiagent Reinforcement Learning in Partially Observable Environments

... and a Plea to "Go Wide"

Marc Lanctot

Joint work with many, many collaborators!

Joint Work with Many Collaborators!

Outline

- **Part 1** [30 min]: Game-Theoretic Approaches to Multiagent RL in Partially Observable Environments

- **Part 2** [20 min]: Beyond Zero-Sum Games and Beyond Domain-Specific Evaluation

Outline

- **Part 1** [30 min]: Game-Theoretic Approaches to Multiagent RL in Partially Observable Environments

- **Part 2** [20 min]: Beyond Zero-Sum Games and Beyond Domain-Specific Evaluation

- Part 1: 45 min
- Part 2: 10 min
- Questions: 5 min

:)

DeepMind

1 Game-Theoretic Approaches to MARL in Partially Observable Games

Normal Form Games: Algorithms

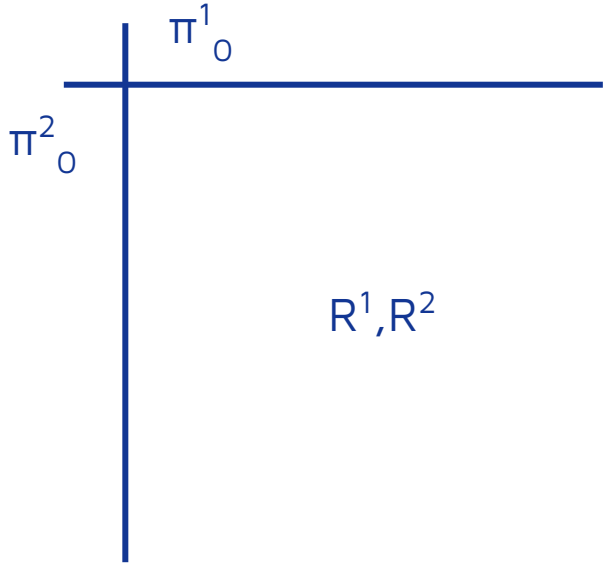
- Inspired by two-player zero-sum games

Two main “streams”

1. Fictitious play / best response stream
2. No-regret stream

Normal Form Games: Algorithms

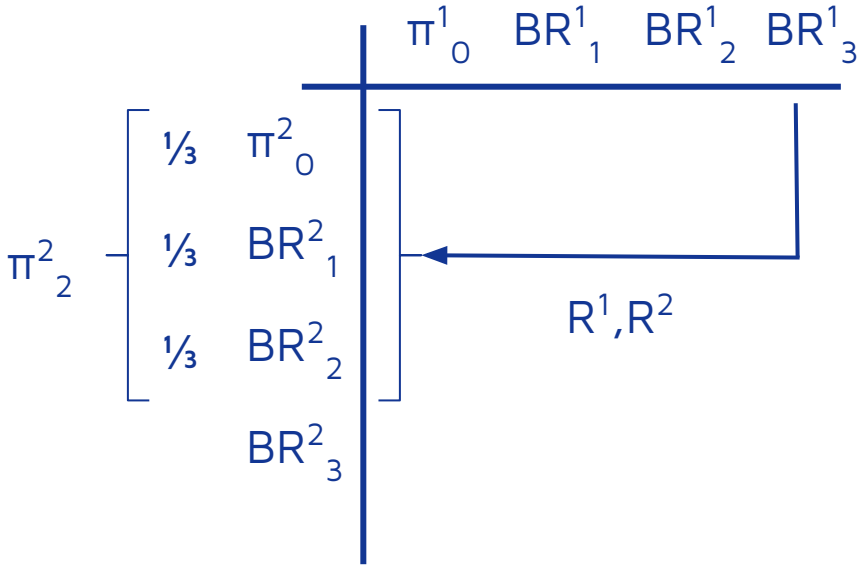
- Fictitious Play:



- Start with an arbitrary policy per player (π^1_0, π^2_0),

Normal Form Games: Algorithms

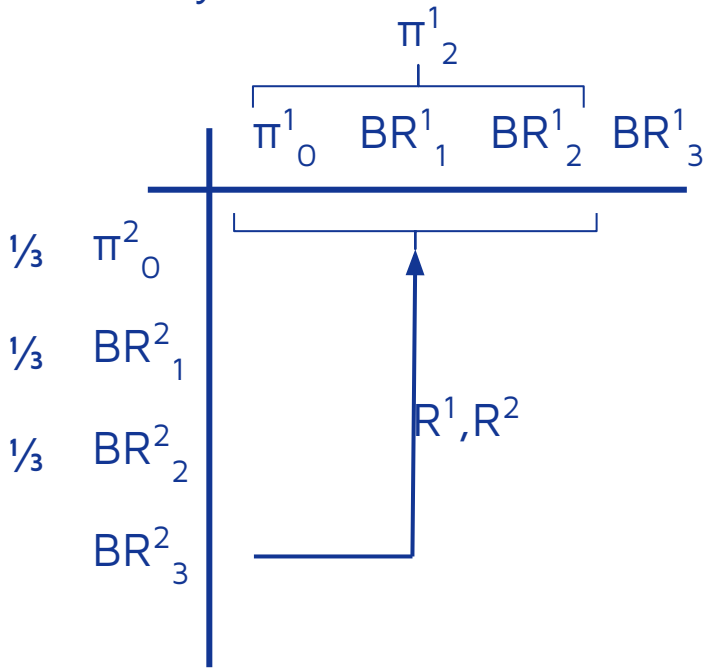
- Fictitious Play:



- Start with an arbitrary policy per player (π^1_0, π^2_0),
  - Then, play best response against a uniform distribution over the past policies of the opponent ($BR^{-i}_{1\,..\,n-1}$).

Normal Form Games: Algorithms

- Fictitious Play:



- Start with an arbitrary policy per player (π^1_0,π^2_0),
  - Then, play best response against a uniform distribution over the past policies of the opponent ($BR^{-i}_{1\,..\,n-1}$).

Normal Form Games: Algorithms

- Fictitious Play:

	R
R	0

(showing row player's utility)

- Start with (R, P, S)= (1, 0, 0), (1, 0, 0)

Normal Form Games: Algorithms

- Fictitious Play:

	R	P
R	0	-1
P	1	0

(showing row player's utility)

- Start with (R, P, S)= (1, 0, 0), (1, 0, 0)
- Iteration 1:
  - BR^1_1, BR^2_1 = P, P
  - (½, ½, 0), (½, ½, 0)

Normal Form Games: Algorithms

- Fictitious Play:

	R	P	P
R	0	-1	-1
P	1	0	0
P	1	0	0

(showing row player's utility)

- Start with (R, P, S)= (1, 0, 0), (1, 0, 0)
- Iteration 1:
  - BR^1_1, BR^2_1 = P, P
  - (½, ½, 0), (½, ½, 0)
- Iteration 2:
  - BR^1_2, BR^2_2 = P, P
  - (⅓, ⅔, 0), (⅓, ⅔, 0)

Normal Form Games: Algorithms

- Fictitious Play:

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

(showing row player's utility)

- Start with (R, P, S)= (1, 0, 0), (1, 0, 0)
- Iteration 1:
  - BR^1_1, BR^2_1 = P, P
  - (½, ½, 0), (½, ½, 0)
- Iteration 2:
  - BR^1_2, BR^2_2 = P, P
  - (⅓, ⅔, 0), (⅓, ⅔, 0)
- Iteration 3:
  - BR^1_3, BR^2_3 = S, S
  - (¼,½,¼), (¼,½,¼)

Normal Form Games: Algorithms

- Fictitious Play:

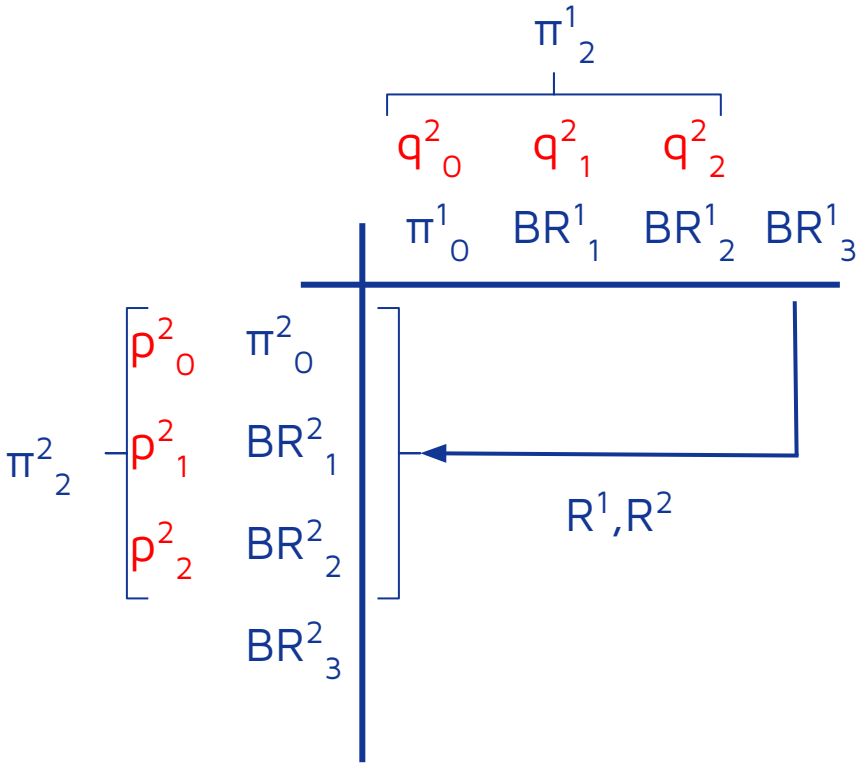
	R	P	P	S	S
R	0	-1	-1	1	1
P	1	0	0	-1	-1
P	1	0	0	-1	-1
S	-1	1	1	0	0
S	-1	1	1	0	0

(showing row player's utility)

- Start with (R, P, S)= (1, 0, 0), (1, 0, 0)
- Iteration 1:
  - BR^1_1, BR^2_1 = P, P
  - (½, ½, 0), (½, ½, 0)
- Iteration 2:
  - BR^1_2, BR^2_2 = P, P
  - (⅓, ⅔, 0), (⅓, ⅔, 0)
- Iteration 3:
  - BR^1_3, BR^2_3 = S, S
  - (¼,½,¼), (¼,½,¼)

Normal Form Games: Algorithms

- double oracle [HB McMahan 2003]:



- Start with an arbitrary policy per player (π^1_0,π^2_0),
  - Compute (p^n,q^n) by solving the game at iteration n
  - Then, best response against (p^n,q^n) and get a new best response (BR^1_n,BR^1_n).

Normal Form Games: Algorithms

- double oracle:

- Start with (R, P, S)= (1, 0, 0), (1, 0, 0)

	R
R	0

(showing row player's utility)

Normal Form Games: Algorithms

- double oracle:

	R	P
R	0	-1
P	1	0

(showing row player's utility)

- Start with (R, P, S)= (1, 0, 0), (1, 0, 0)
- Iteration 1:
  - BR^1_1, BR^2_1 = P, P
  - Solve the game : (0, 1, 0), (0, 1, 0)

Normal Form Games: Algorithms

- double oracle:

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

(showing row player's utility)

- Start with (R, P, S)= (1, 0, 0), (1, 0, 0)
- Iteration 1:
  - BR^1_1, BR^2_1 = P, P
  - Solve the game : (0, 1, 0), (0, 1, 0)
- Iteration 2:
  - BR^1_2, BR^2_2 = S, S
  - (⅓, ⅓, ⅓), (⅓, ⅓, ⅓)

Normal Form Games: Algorithms

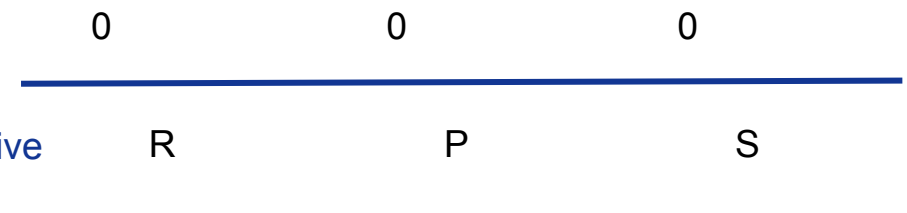
- Regret-matching (Hart & Mas-Colell '00):

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

(showing row player's utility)

Normal Form Games: Algorithms

- Regret-matching (Hart & Mas-Colell '00):
  - For row player 1, column player **fixed**



0.2 0.4 0.4

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

(showing row player's utility)

Normal Form Games: Algorithms

- Regret-matching (Hart & Mas-Colell '00):
  - For row player 1, column player **fixed**
  - t=0, π^1_0 = (⅓, ⅓, ⅓), R^1 = 0

	0.2	0.4	0.4
	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

(showing row player's utility)

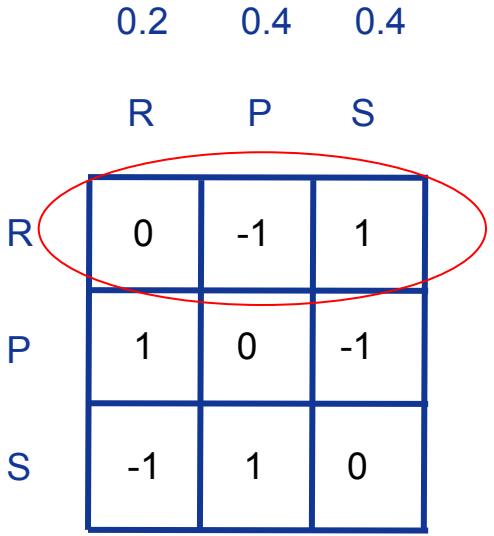
	R	P	S
Cumulative Regrets	0	0	0

Normal Form Games: Algorithms

- Regret-matching (Hart & Mas-Colell '00):
  - For row player 1, column player **fixed**
  - $t=0$, $\pi^1_0 = (⅓, ⅓, ⅓)$, $R^1 = 0$
  - Reg. *not* playing rock:
    - $= (-0.4 + 0.4) - 0 = 0$

Cumulative Regrets

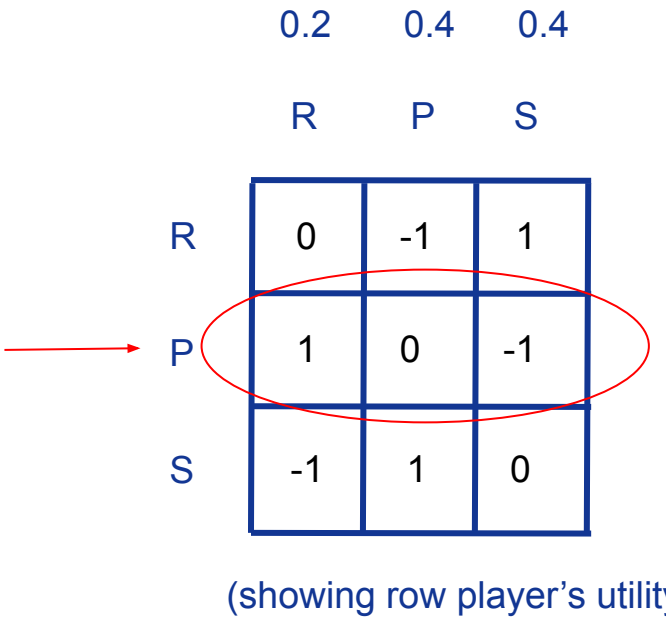
R P S



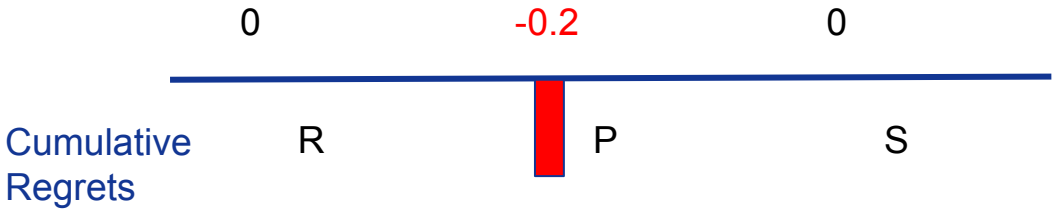
(showing row player's utility)

Normal Form Games: Algorithms

- Regret-matching (Hart & Mas-Colell '00):
  - For row player 1, column player **fixed**
  - $t=0$, $\pi^1_0 = (⅓, ⅓, ⅓)$, $R^1 = 0$
  - Reg. *not* playing paper:
    - = (0.2 - 0.4) - 0 = -0.2

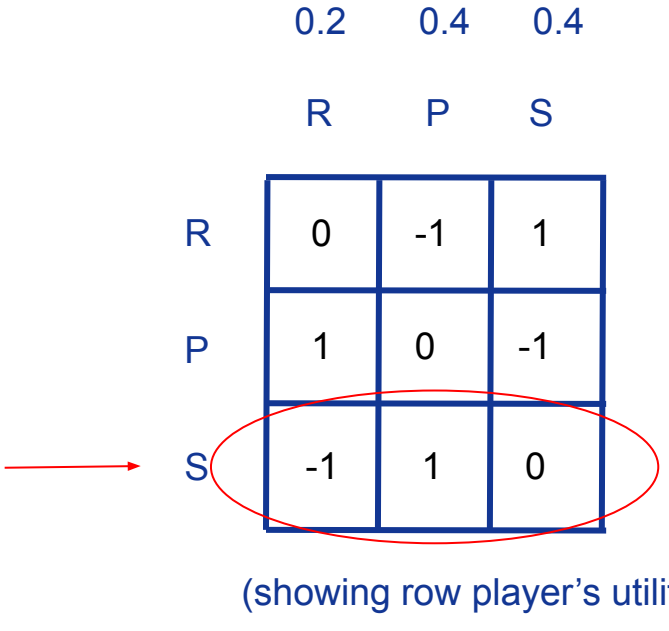


(showing row player's utility)

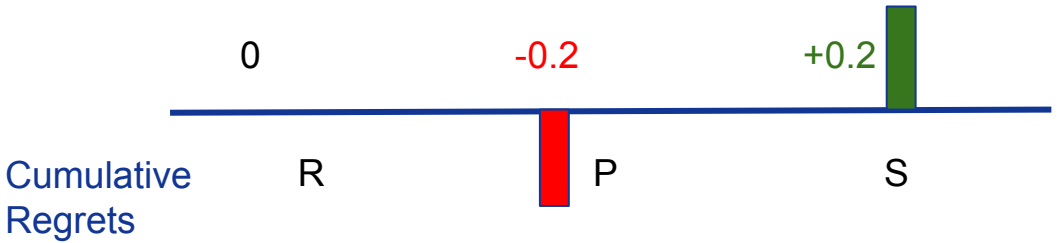


Normal Form Games: Algorithms

- Regret-matching (Hart & Mas-Colell '00):
  - For row player 1, column player **fixed**
  - $t=0$, $\pi^1_0 = (⅓, ⅓, ⅓)$, $R^1 = 0$
  - Reg. *not* playing scissors:
    - $= (-0.2 + 0.4) - 0 = +0.2$



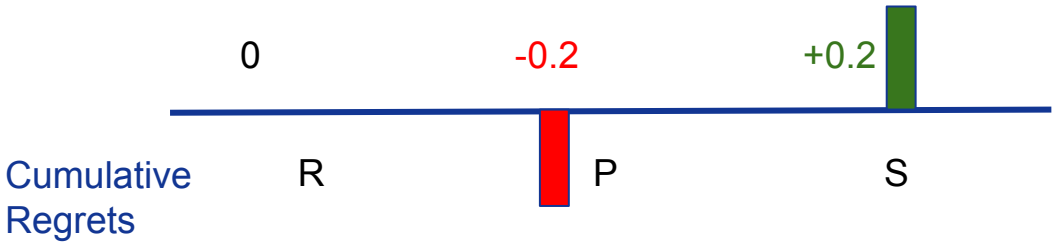
(showing row player's utility)



Normal Form Games: Algorithms

- Regret-matching (Hart & Mas-Colell '00):
  - For row player 1, column player **fixed**
  - t=1, π^1_1 = (0, 0, 1)

Normalize positive cumulative regrets (set others to 0)



	0.2 R	0.4 P	0.4 S
R	0	-1	1
P	1	0	-1
S	-1	1	0

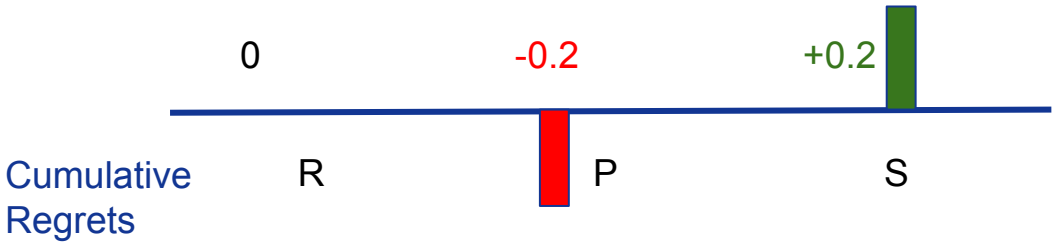
(showing row player's utility)

Normal Form Games: Algorithms

- Regret-matching (Hart & Mas-Colell '00):
  - For row player 1, column player **fixed**
  - t=1, π^1_1 = (0, 0, 1), R^1 = 0.2

	0.2	0.4	0.4
	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

(showing row player's utility)

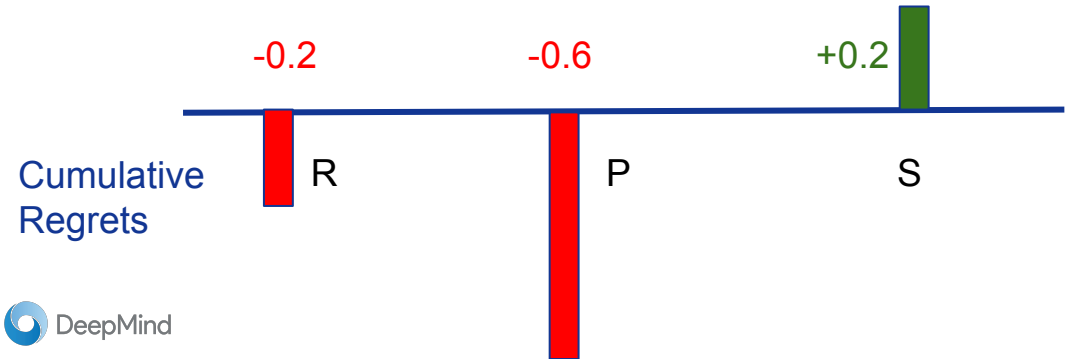


Normal Form Games: Algorithms

- Regret-matching (Hart & Mas-Colell '00):
  - For row player 1, column player **fixed**
  - t=2, π^1_2 = (0, 0, 1), R^1 = 0.2

	0.2 R	0.4 P	0.4 S
R	0	-1	1
P	1	0	-1
S	-1	1	0

(showing row player's utility)

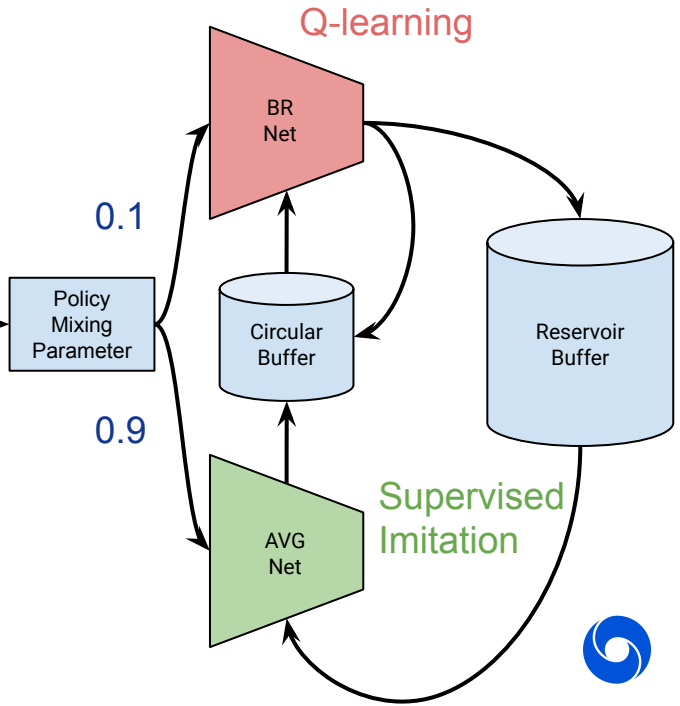


Fictitious Self-Play (FSP) [Heinrich, Lanctot, & Silver '15]

- **Idea:** Fictitious play + reinforcement learning in one online agent
- Update rule in sequential setting *equivalent* to standard fictitious play (matrix game)

1. **Best response (BR):**
   - Estimate a best response
   - Trained via RL (e.g. Q-learning)
   - Circular buffer of (s, a, s', r) tuples

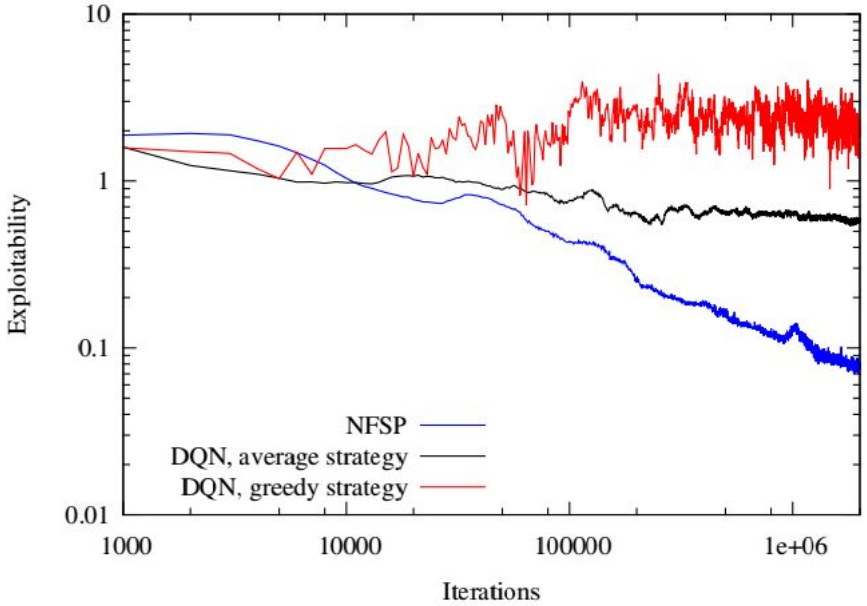
2. **Average policy (AVG):**
   - Estimate the time-average policy
   - Trained via supervised learning
   - Reservoir buffer of (s, a) pairs



Neural Fictitious Self-Play (NFSP) [Heinrich & Silver '16]

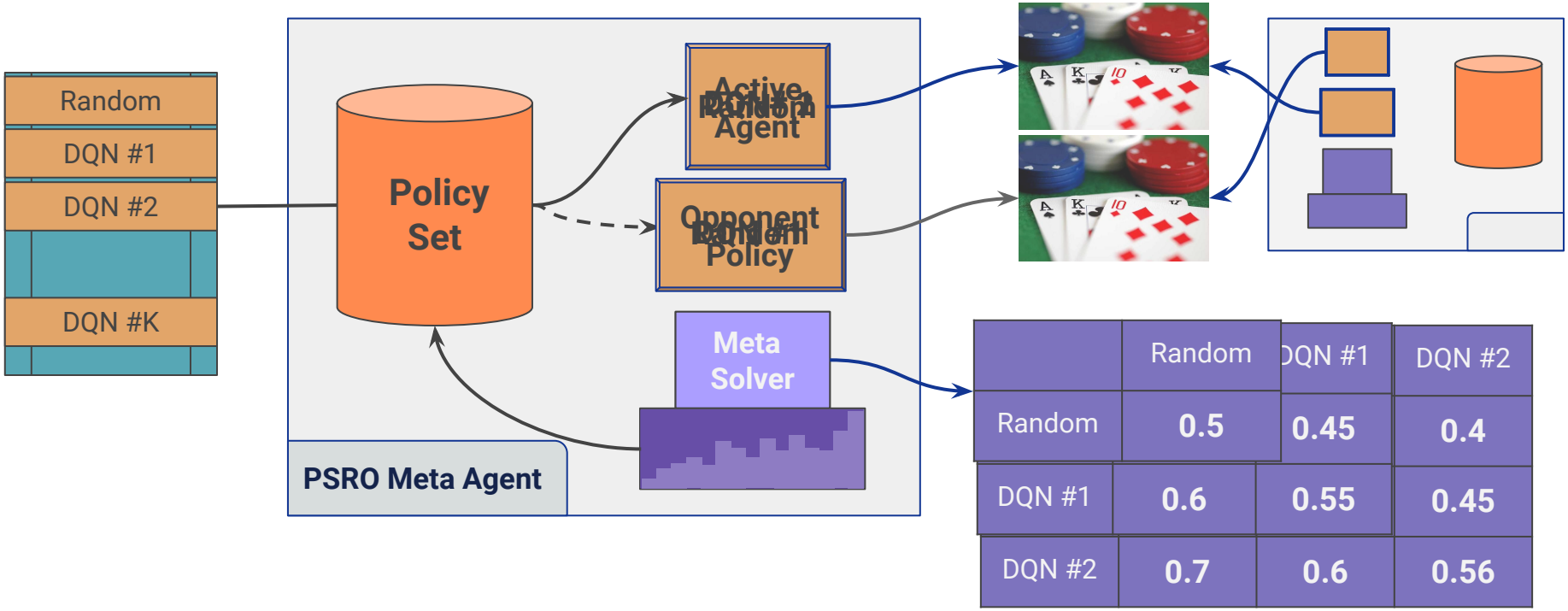
- Approximate NE via two neural networks:
- Leduc Hold'em poker experiments:

"Closeness" to Nash



- Competitive with strong computer poker programs when it was released

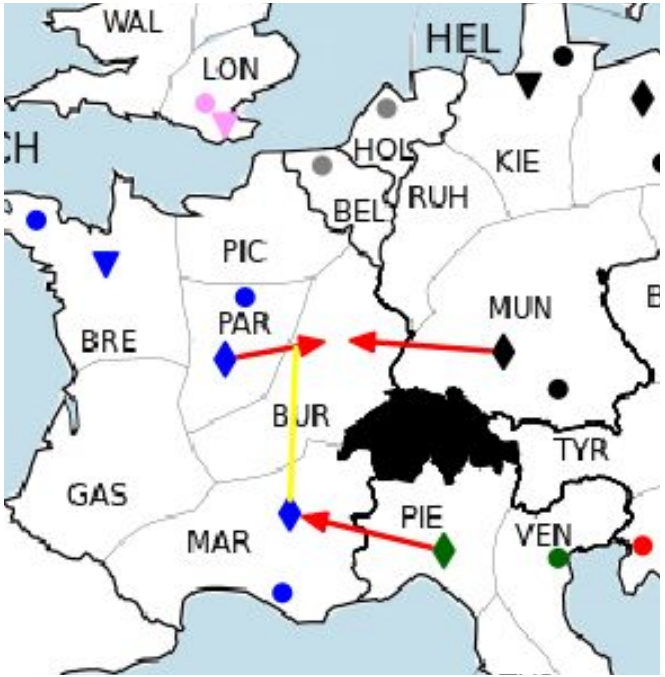
Policy-Space Response Oracles (Lanctot et al. '17)



	Random	DQN #1	DQN #2
Random	0.5	0.45	0.4
DQN #1	0.6	0.55	0.45
DQN #2	0.7	0.6	0.56

Best Response Policy Iteration and Diplomacy
[Anthony et al. '20]

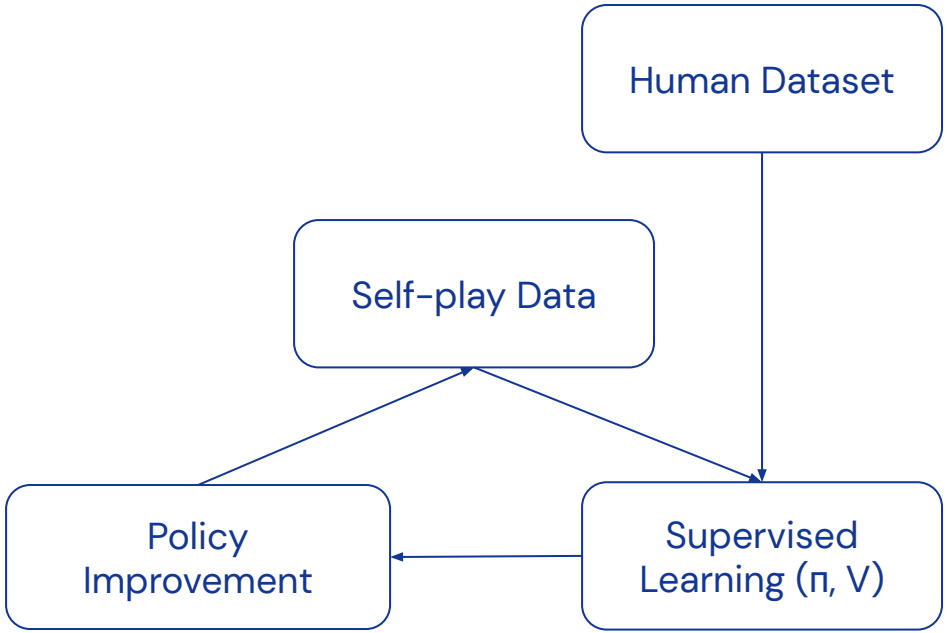
- Classic board game
- 7-player game
- Simultaneous moves
- 10^{21} - 10^{64} legal actions *per turn*
- Mixed-motives:
  - Winning requires alliances
  - Players negotiate for territory



Current focus on *no-press variant*.

Best-Response Policy Iteration [Anthony et al. ‘20]

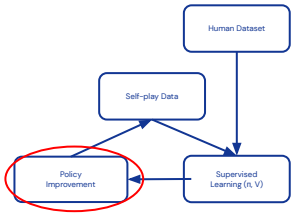
1. Starting point: Human imitation
2. Policy Improvement Operator
3. Generate and imitate new self-play data with improved policies



DipNet (Paquette et al. ‘19)

Best-Response Policy Iteration [Anthony et al. '20]

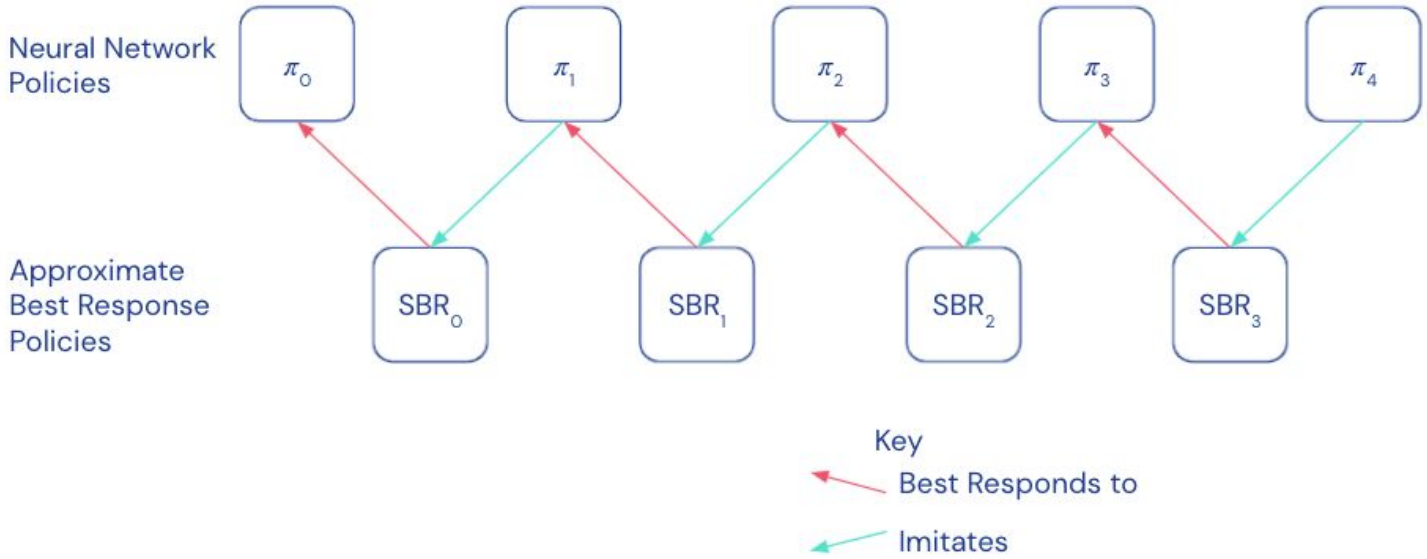
- Input:
  - Base policy π_b
  - Candidate policy π_c
  - Environment dynamics *T(s, a) -> s'*
  - Value function *V(s')*
- Algorithm:
  - At each turn, given state *s*:
    - Sample a few base profiles a_{-i} from $\pi_b(s)$ for all players but *i*
    - Sample several candidate actions a_i from $\pi_c(s)$
    - Plug the sampled actions into $T(s, a_i, a_{-i}) \rightarrow s'$
    - Get *V(s')* (for player *i*)
    - Play the candidate action with the best average value against the base profiles → **sampled best response** (SBR)



BRPI Policy Improvement

But what best response should we be imitating?

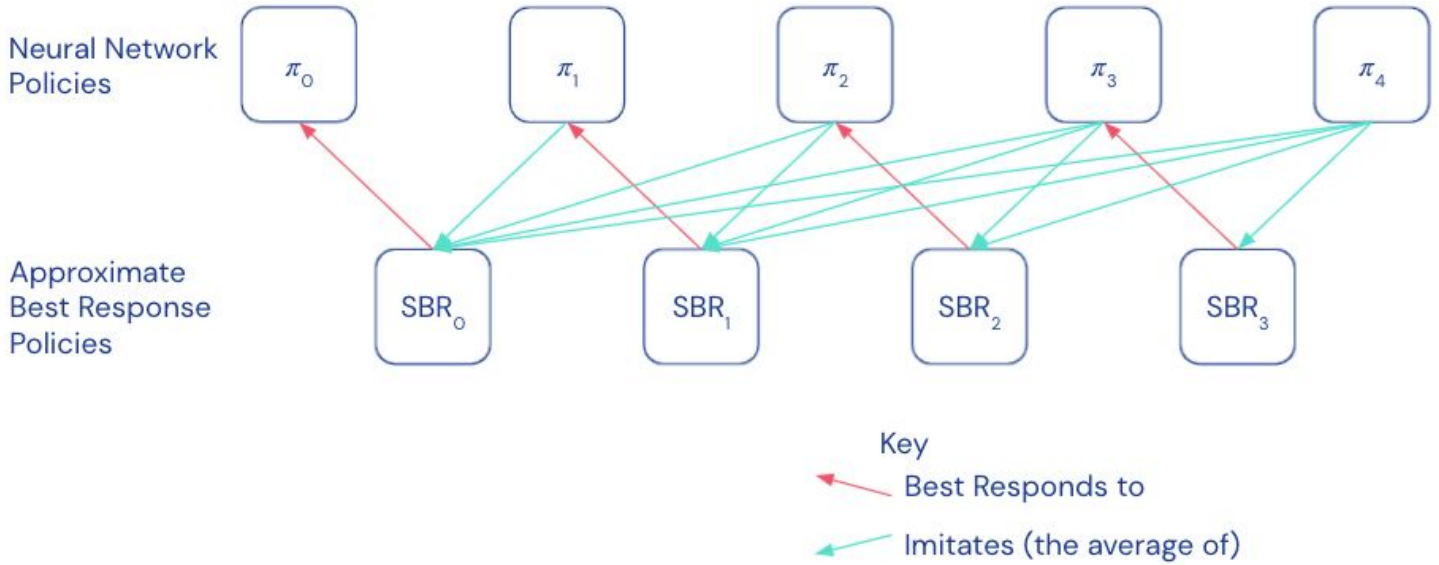
- Iterated Best Responses



BRPI Policy Improvement

But what best response should we be imitating?

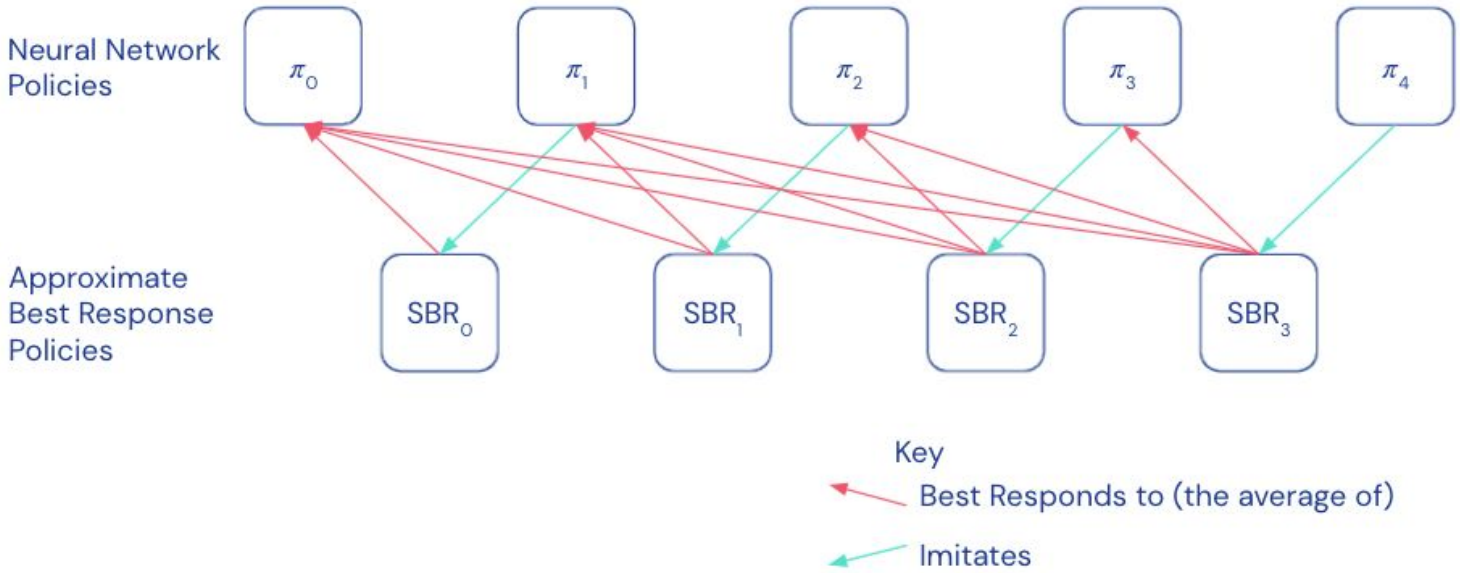
- Iterated Best Responses
- Fictitious Play (1) -- “à la NFSP”



BRPI Policy Improvement

But what best response should we be imitating?

- Iterated Best Responses
- Fictitious Play (1) -- “à la NFSP”
- Fictitious Play (2)



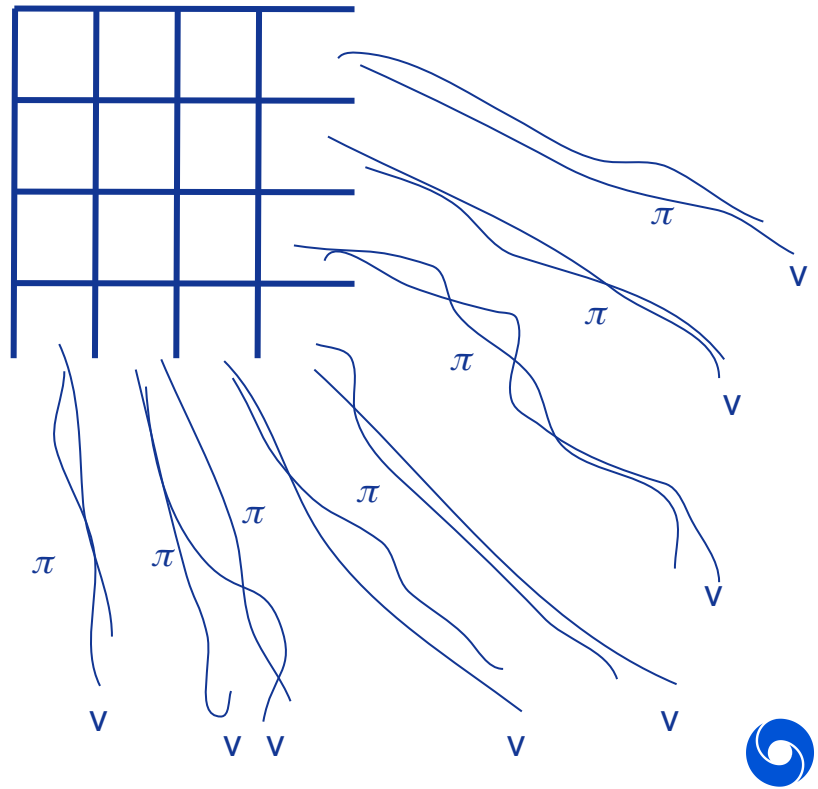
BRPI in Diplomacy: Results

	SL [90]	A2C [90]	SL (ours)	FPPI-1	IBR	FPPI-2
SL [90]	14.2%	8.3%	16.3%	2.3%	1.8%	*0.8%*
A2C [90]	15.1%	14.2%	15.3%	2.3%	1.7%	*0.9%*
SL (ours)	12.6%	7.7%	14.1%	3.0%	1.9%	*1.1%*
FPPI-1	**26.4%**	28.0%	**25.9%**	14.4%	7.4%	*4.5%*
IBR	20.7%	30.5%	**25.8%**	20.3%	12.9%	*10.9%*
FPPI-2	19.4%	**32.5%**	20.8%	**22.4%**	**13.8%**	***12.7%***

Table 1: Average scores for 1 row player vs 6 column players. BRPI methods give an improvement over A2C or supervised learning. All numbers accurate to a 95% confidence interval of $\pm 0.5\%$. Bold numbers are the best value for single agents against a given set of 6 agents, italics are for the best result for a set of 6-agents against each single agent.

Human-Level No-Press Diplomacy [Gray et al. '20]

- Human data → DipNet
- DipNet provides policy π and value net v
- Use regret matching in stage game
- Get payoffs from sims / search
  - Use policy for rollouts + selection
  - Value net after some horizon
- Human-level play on webdiplomacy



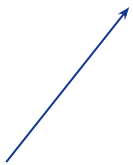
Regret Minimization

Counterfactual regret minimization (CFR) (Zinkevich et al. '08):

Basis of success in Poker AI for two-player zero-sum games:

Player 1 π_1^0

Player 2 π_2^0



Initial policies iteration, t = 0

Regret Minimization

Counterfactual regret minimization (CFR) (Zinkevich et al. '08):

Basis of success in Poker AI for two-player zero-sum games:

Player 1 $\pi_1^0 \longrightarrow \pi_1^1$

Player 2 $\pi_2^0 \longrightarrow \pi_2^1$

Regret Minimization

Counterfactual regret minimization (CFR) (Zinkevich et al. '08):

Basis of success in Poker AI for two-player zero-sum games:

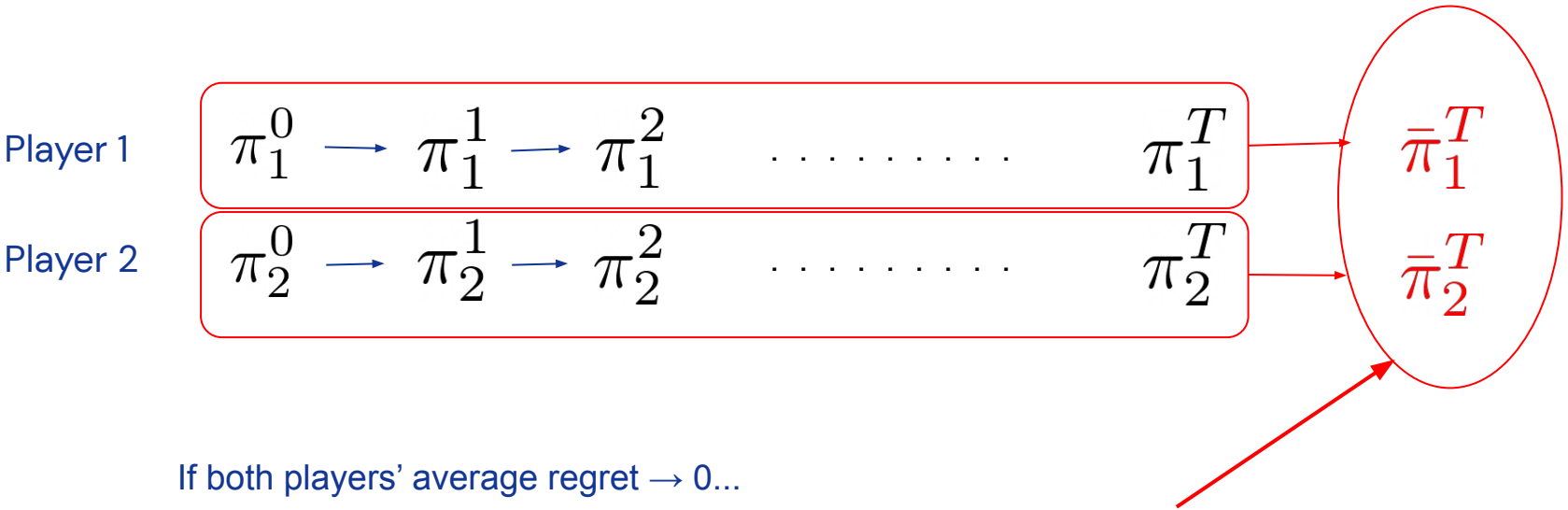
Player 1 $\pi_1^0 \longrightarrow \pi_1^1 \longrightarrow \pi_1^2$

Player 2 $\pi_2^0 \longrightarrow \pi_2^1 \longrightarrow \pi_2^2$

Regret Minimization

Counterfactual regret minimization (CFR) (Zinkevich et al. '08):

Basis of success in Poker AI for two-player zero-sum games:



If both players' average regret → 0...

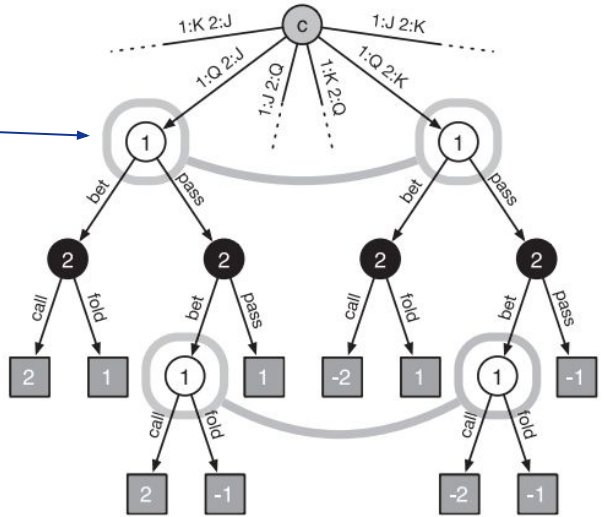
Average strategy (approx Nash in two-player zero-sum)

Counterfactual Regret Minimization (CFR)

[Zinkevich et. al' 08]

- Tabular method (policy iteration)
- Each *information state*, s:
  - Compute **counterfactual regrets** r(s,a)
  - Accumulate: R(s,a) += r(s,a)
  - Use regret-matching for new π(s)

e.g.



Advantage vs. Regrets

A key notion in CFR is an **immediate regret**:

$$r(s,a) = q^c_{\pi,i}(s,a) - v^c_{\pi,i}(s)$$

counterfactual q-value

joint policy

return to player i

(player to play at s)

→ This is just a (counterfactual) advantage!

RL values vs. Counterfactual values

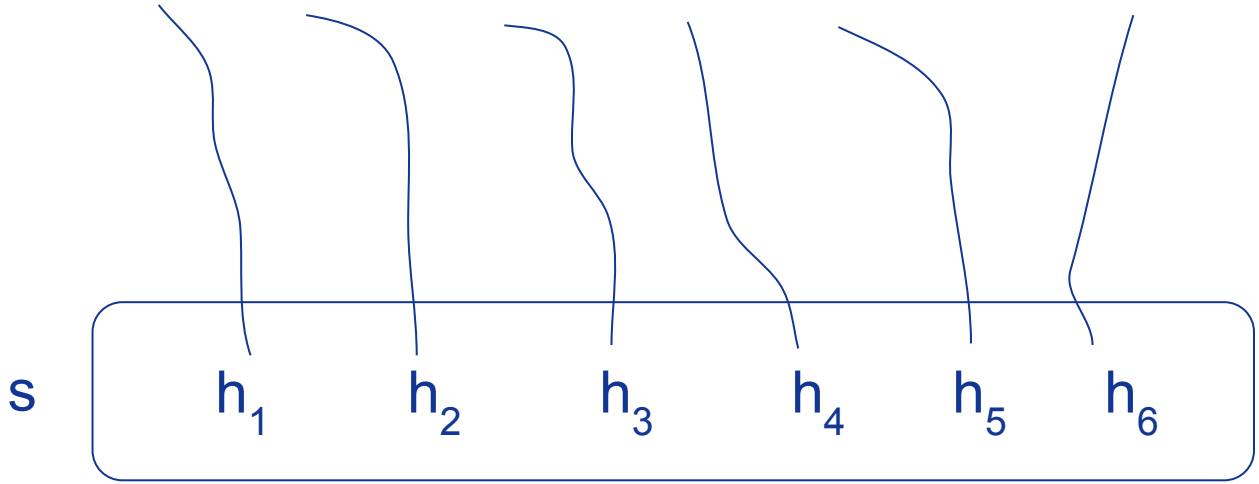
$$q_{\pi,i}(s,a) = \frac{1}{\beta_{-i}(s)} q^{c}_{\pi,i}(s,a)$$

"RL-style" q-value (conditioned on reaching s)

Probability that s is reached given opponents' policies

counterfactual q-value (weighted sum over histories)

Bayes Normalizer



$$\beta_{-i}(s, \pi) = \sum_{h \in s} \Pr(h)$$

Q-based Policy Gradient

A.K.A. “all-actions” policy gradient

A.K.A. Mean Actor-Critic (Allen et al. ‘17)

$$\nabla_{\boldsymbol{\theta}}^{\mathrm{QPG}}(s) = \sum_a [\nabla_\theta \pi(s, a; \boldsymbol{\theta})] \left(q(s, a; \mathbf{w}) - \sum_b \pi(s, b; \boldsymbol{\theta}) q(s, b, \mathbf{w}) \right)$$

Regret-based Policy Gradient [Srinivasan et al. '18]

Instead of maximizing objective, **minimize regret**:

$$\nabla_{\boldsymbol{\theta}}^{\mathrm{RPG}}(s) = -\sum_{a} \nabla_{\theta} \left(q(s,a;\mathbf{w}) - \sum_{b} \pi(s,b;\boldsymbol{\theta}) q(s,b;\mathbf{w}) \right)^{+}$$

where $(x)^{+} = \max(0, x)$

→ Gradient **descent** (instead of ascent)

Replicator Dynamics (Taylor & Jonker '78)

- Population state x evolves inspired by biologically inspired operators
- Proportion of member i, x_i , grows according to their fitness f

$$\dot{x}_i = x_i \left(f(\boldsymbol{x})_i - \bar{f}(\boldsymbol{x}) \right)$$

$$\bar{f}(\boldsymbol{x}) = \sum_j x_j f(\boldsymbol{x})_j$$

in **Reinforcement Learning** terms:

$$\dot{\pi}(a) = \pi(a)[q(a) - v]$$

Policy Gradient vs. Replicator Dynamics

Policy Gradient (Advantage Actor-Critic)

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\pi}} \left[\nabla_{\boldsymbol{\theta}} \log \pi(a_t \mid s_t; \boldsymbol{\theta}) A(s_t, a_t; \boldsymbol{w}, \boldsymbol{\theta}) \right]$$

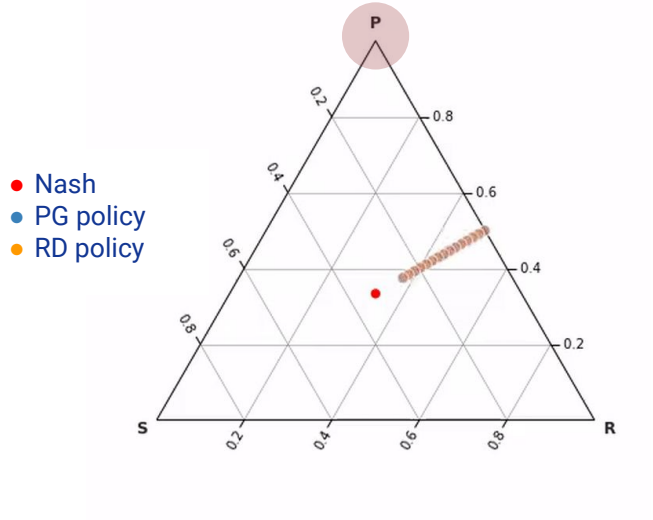
Replicator Dynamics

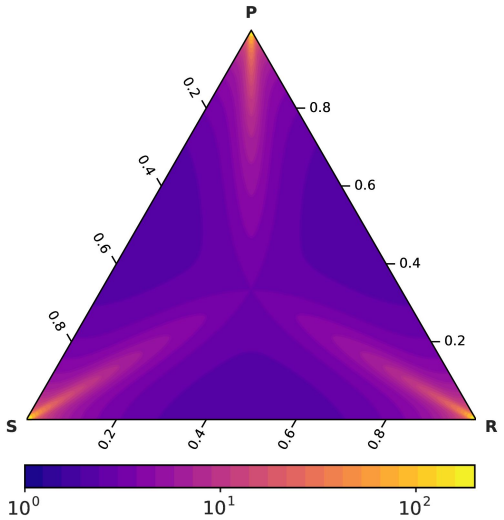
$$\dot{\pi}(a) = \pi(a) A(a)$$

logit space $\boldsymbol{\pi} = softmax(\boldsymbol{y})$ stateless tabular case

$$y_t(a) = y_{t-1}(a) + \eta \pi(a) A(a)$$

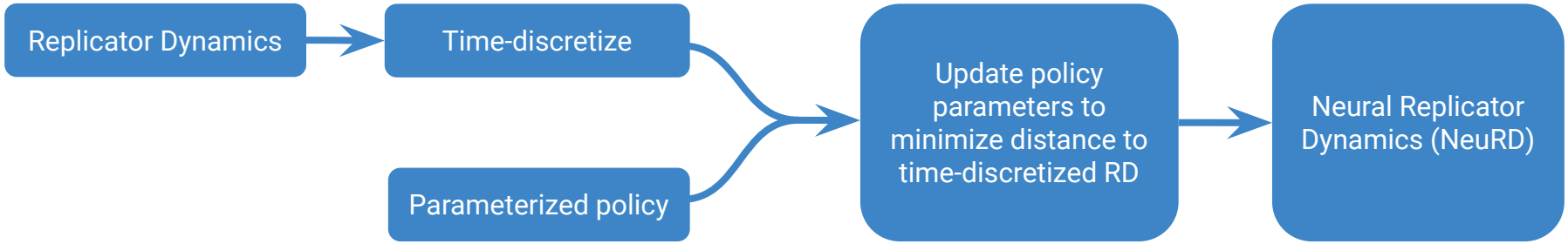
$$y_t(a) = y_{t-1}(a) + \eta A(a)$$





$$\frac{||\dot{\boldsymbol{\pi}}_{\mathrm{RD}}||}{||\dot{\boldsymbol{\pi}}_{\mathrm{PG}}||}$$

Neural Replicator Dynamics [Omidshafiei, Hennes, Morrill et al. '19]

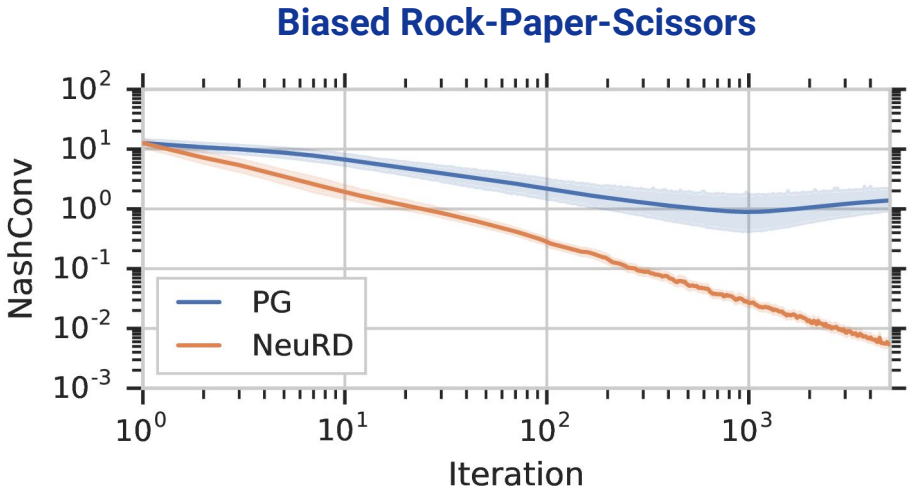


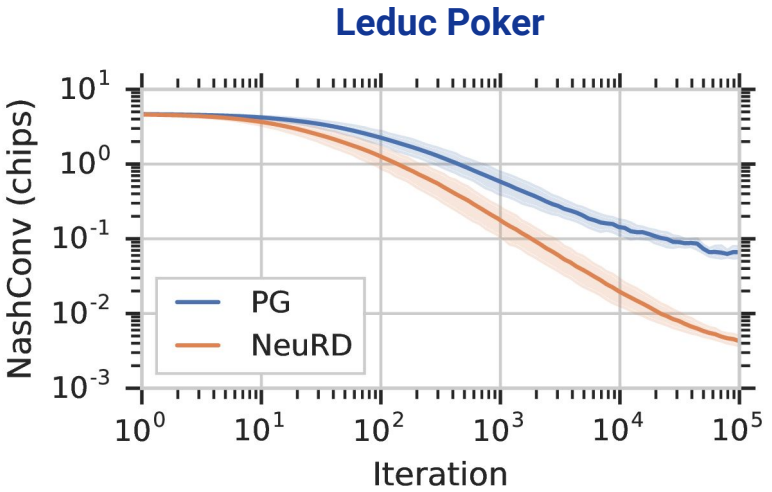
$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t+1} + \eta \sum_{s,a} \nabla_{\boldsymbol{\theta}} y_{t-1}(s_t, a_t; \boldsymbol{\theta}) A(s_t, a_t; \boldsymbol{\theta}, \boldsymbol{w})$$

Logits, where policy is $\boldsymbol{\pi} = softmax(\boldsymbol{y})$

Advantage q(s,a)-v(s)

Neural Replicator Dynamics: Results

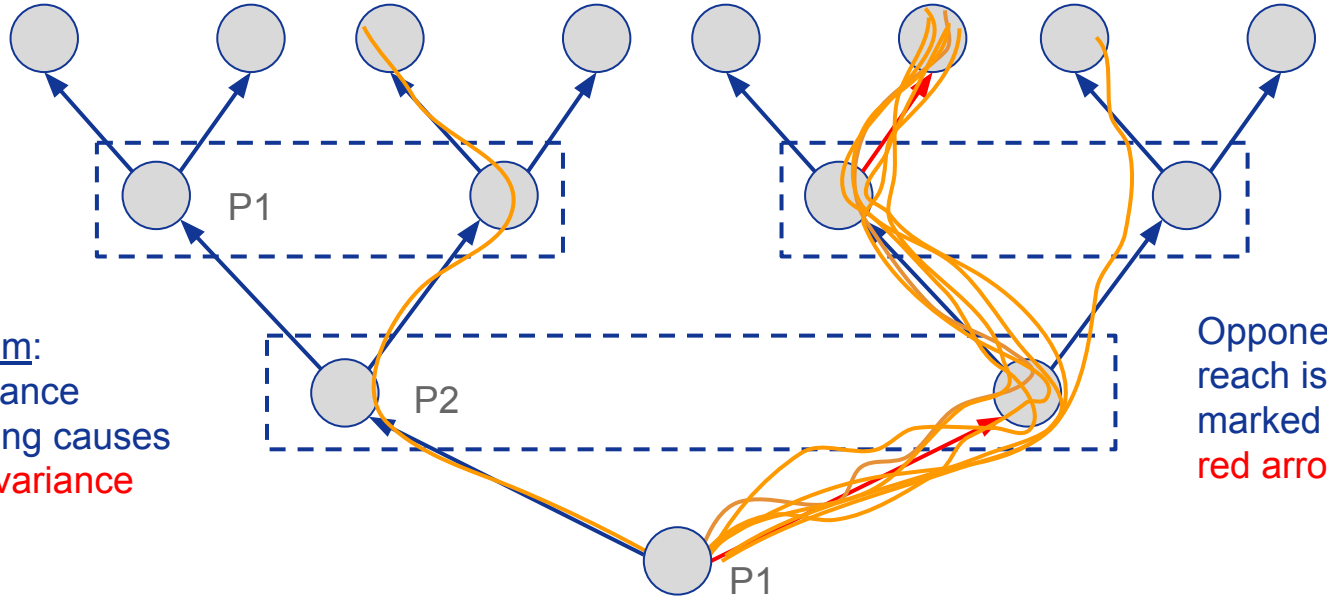




Advantage Regret-Matching Actor-Critic [Gruslys et al. '20]

Goal: sample-based model-free CFR with function approximation.

Monte Carlo CFR [Lanctot et al. '09]: Sample trajectories in interesting portions of the game tree.



Problem: Importance sampling causes **huge** variance

Opponent reach is marked with red arrows

Limitations of (Outcome Sampling) Monte Carlo CFR [Lanctot et al '09]

Memory hungry
tabulates all information states

No generalization

Huge variance

We want to solve all those problems by using neural networks and RL-style trajectory sampling

Problem 1: Cumulative Regrets

Problem: neural networks can not accumulate regrets

Solution: reformulate CFR in terms of **mean regrets**

Instead of cumulative regrets $R^T(s,a)$

Learn an estimate $\hat{R}^T(s,a) \rightarrow \frac{R^T(s,a)}{T}$

Inspired by Regression CFR (RCFR) [Waugh et al. '15]

Problem 2: Sampling and Variance

Corresponds to T CFR iterations

Problem: MCCFR estimator of $\hat{R}^T(s,a)$ can have huge variance

Solution:

1. Maintain a **reservoir of past joint policies** over **epochs** { 1, 2, …, T }
2. At current **epoch** t, generate data:
   a. Sample past checkpoint j ~ { 1… t-1 } uniformly
   b. Sample our actions with exploratory behavior policy $u_i^t(s)$
   c. Sample opponent actions using π^j
   d. Tabulate sampled regrets $\hat{r}^j(s,a)$ with **history-based critics**
3. Train regressor to predict mean regrets

Problem 2: Sampling and Variance

Problem: MCCFR estimator of $\hat{R}^T(s,a)$ can have huge variance

Solution:

1. Maintain a **reservoir of past joint policies** over **epochs** { 1, 2, …, T }
2. At current **epoch** t, generate data:
   a. Sample past checkpoint j ~ { 1… t-1 } uniformly
   b. Sample our actions with exploratory behavior policy $u_i^t(s)$
   c. Sample opponent actions using π^j
   d. Tabulate sampled regrets $\hat{r}^j(s,a)$ with **history-based critics**
3. Train regressor to predict mean regrets

As a result, we learn $\hat{W}(s,a) \rightarrow W(s,a) = R(s,a)k(s)$

Information state level constant

Problem 3: History / world state critics

Problem:

We need to evaluate advantages $q_{\pi^j}(h, a) - \sum_b q_{\pi^j}(h, b)$

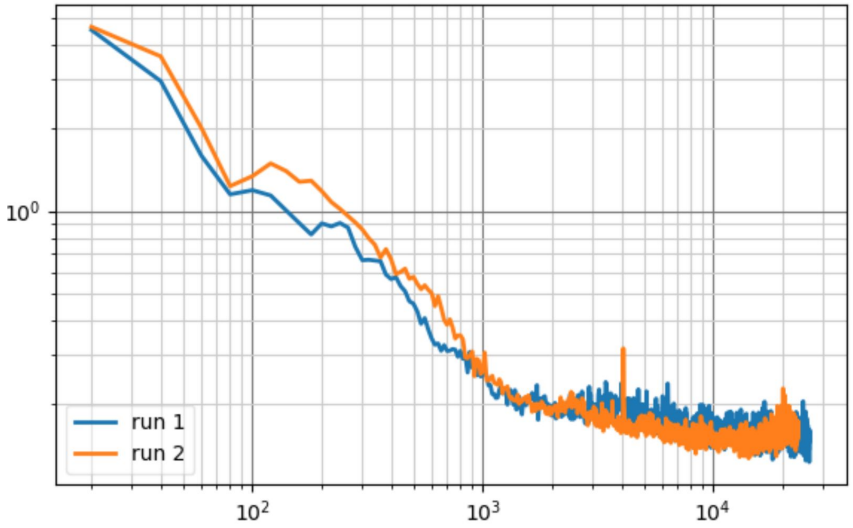
But in epoch T we produce trajectories with an exploratory behavior policy $u_i^t(s)$

Solution:

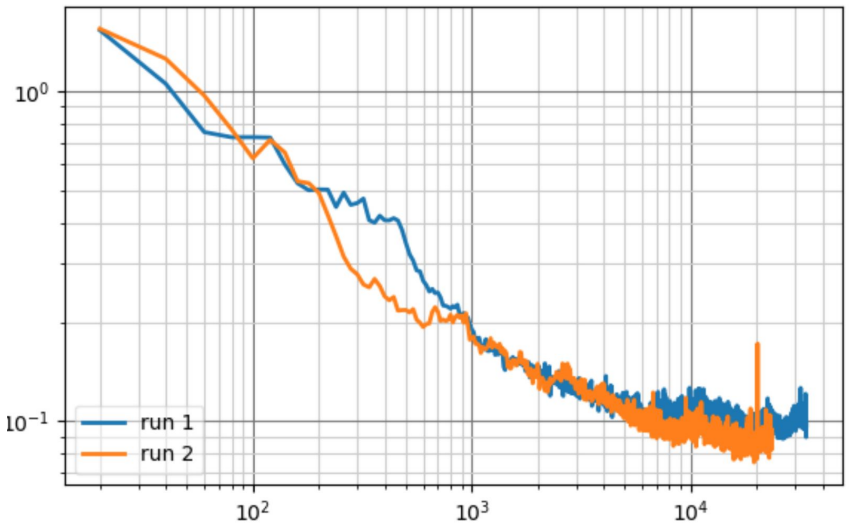
Train value functions using off-policy-RL.

ARMAC Results: benchmark domains

Leduc Poker

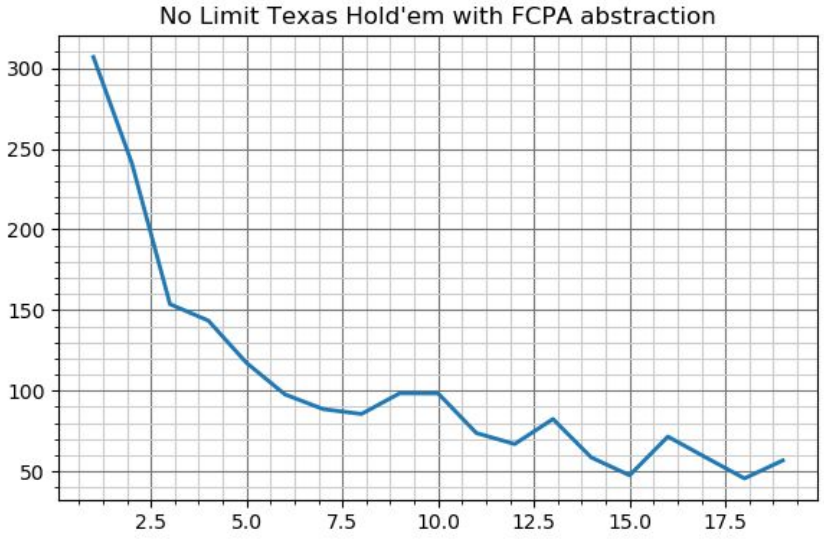


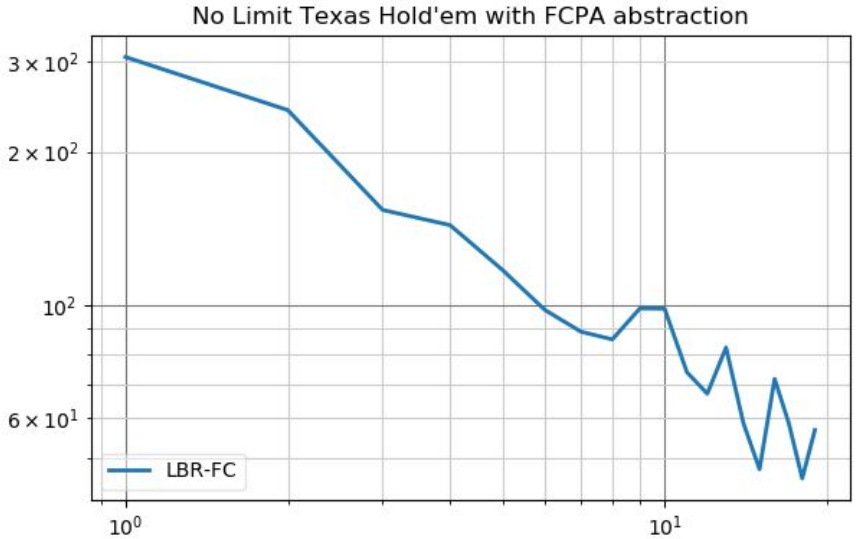
Liars dice



Stable learning curves.

ARMAC Results: action-abstracted no-limit Hold'em



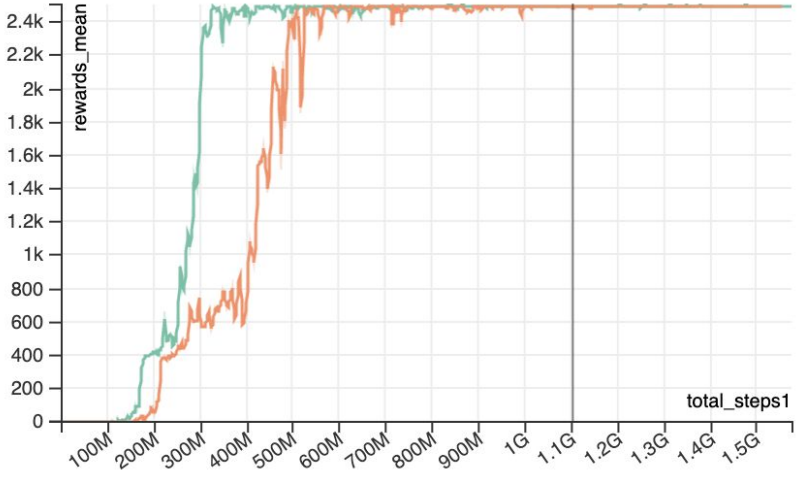


Local Best Response (LBR) bound on exploitability: y-axis = exploitation value, x-axis = epochs
Roughly matches the performance of state of art Poker bots from 2016

ARMAC Results: Atari

Breakout

Montezuma's Revenge



Hypothesis: regret matching has nice exploratory policies

Learning with Regularization: Friction FoReL
[Perolat et al. '21]

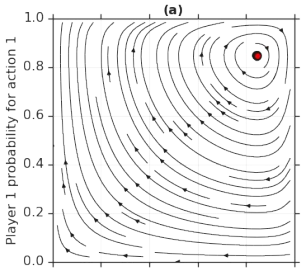
Follow The Regularized Leader:

$$y_t^i(a^i) = \int_0^t Q^i_{\pi_s}(a^i)ds \quad \text{and} \quad \pi_t^i = \text{argmax}_{p\in\Delta A}\Lambda^i(p, y_t^i)$$

With : $\Lambda^i(p,y) = \langle y, p\rangle - \phi_i(p)$ and $\phi_i(p)$ is a regularisation for the policy projection.

In zero-sum two-player games, the following quantity is preserved and the learning trajectory is recurrent:

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

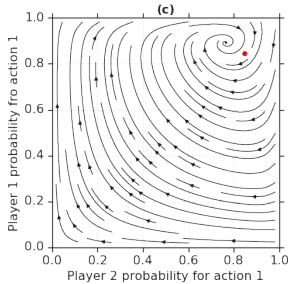


Adding a policy dependent term:

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

This policy dependent term transforms a recurrent learning dynamic to a convergent one:

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

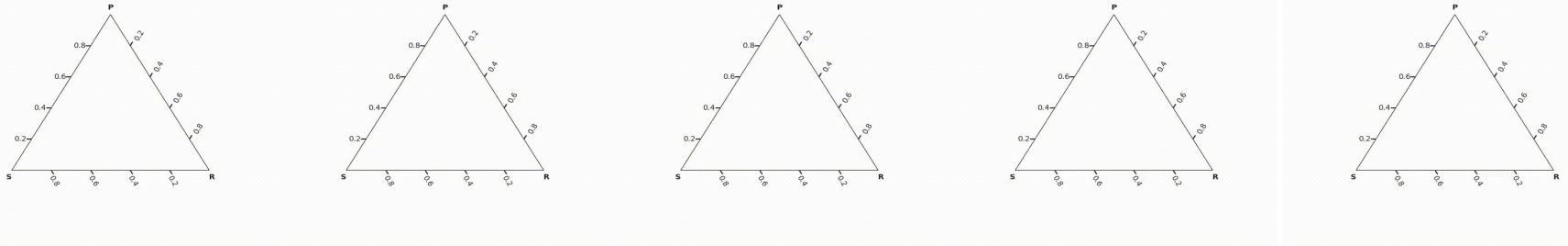


Learning with Regularization Friction FoReL
[Perolat et al. '21]

Video 50x slower

Increasing speed of convergence

No regularization (0.0) | Small regularization (0.05) | Medium regularization (0.1) | Large regularization (0.5) | Absurd regularization (10.0)



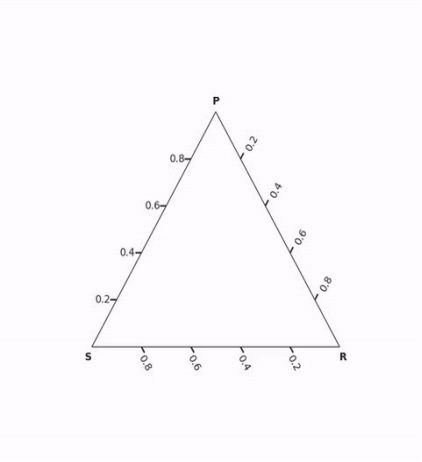
Increasing bias to the solution

We want to learn at high regularization and low bias!

Learning with Regularization: Friction FoReL
[Perolat et al. '21]

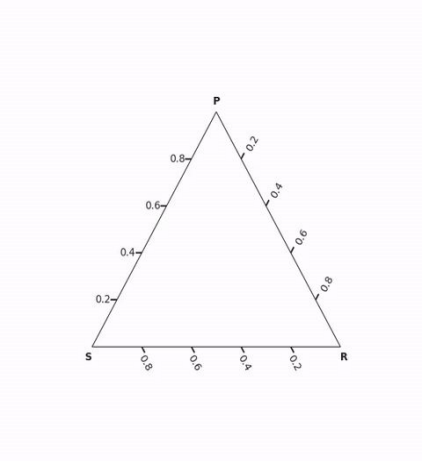
Regularization centered around [⅓, ⅓, ⅓].

π_0



Solution : [0.38, 0.48, 0.12] π_1

Regularization centered around [0.38, 0.48, 0.12]. π_1

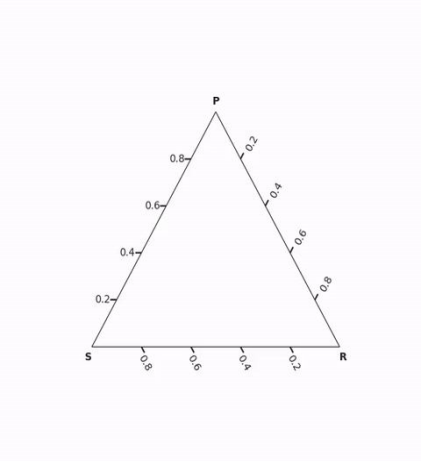


Solution : [0.29, 0.62, 0.07]

Learning with Regularization: Friction FoReL
[Perolat et al. '21]

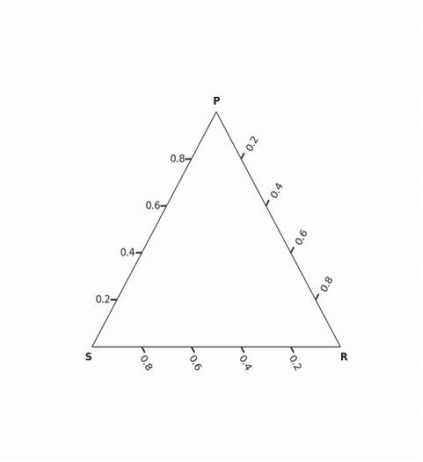
Regularization centered around [⅓, ⅓, ⅓].

π_0



Solution : [0.38, 0.48, 0.12] π_1

Regularization centered around [0.38, 0.48, 0.12]. π_1



Solution : [0.29, 0.62, 0.07]

$$r_\pi^i(a) = r^i(a^i, a^{-i}) - \eta \log \frac{\pi^i(a^i)}{\pi_0^i(a^i)} + \eta \log \frac{\pi^{-i}(a^{-i})}{\pi_0^{-i}(a^{-i})}$$

Converges to π_1

$$r_\pi^i(a) = r^i(a^i, a^{-i}) - \eta \log \frac{\pi^i(a^i)}{\pi_1^i(a^i)} + \eta \log \frac{\pi^{-i}(a^{-i})}{\pi_1^{-i}(a^{-i})}$$

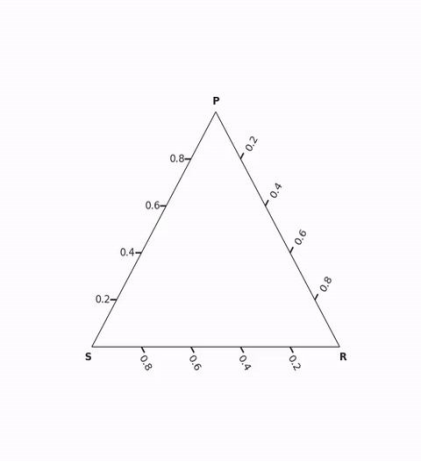
Converges to π_2

Recentering around the previous fixed point will decrease strictly the distance to the Nash of the original game:

$$\Xi(\pi^*, \pi_k) - \Xi(\pi^*, \pi_{k-1}) = \underbrace{-\Xi(\pi_k, \pi_{k-1}) + \frac{1}{\eta}\sum_{i=1}^{N}(m_k^i + \delta_k^i + \kappa_k^i)}_{<0}$$

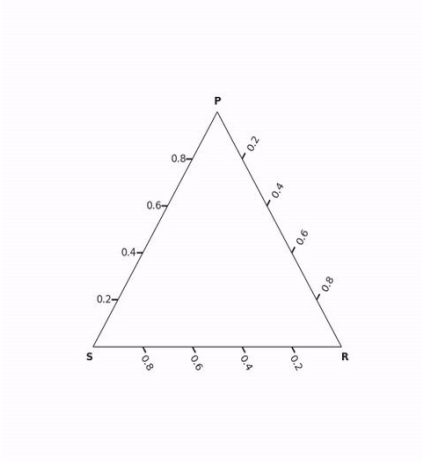
Learning with Regularization

Regularization centered around [⅓, ⅓, ⅓].



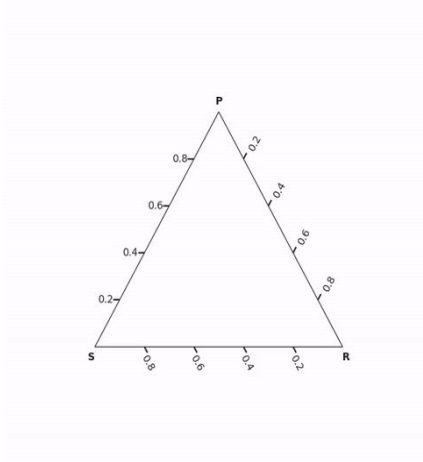
Solution : [0.38, 0.48, 0.12]

Regularization centered around [0.38, 0.48, 0.12].



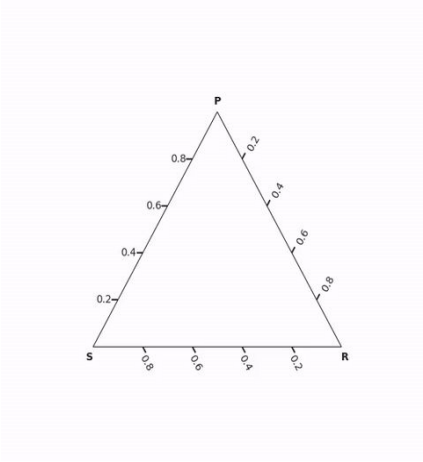
Solution : [0.29, 0.62, 0.07]

Regularization centered around [0.29, 0.62, 0.07].



Solution : [0.19, 0.72, 0.07]

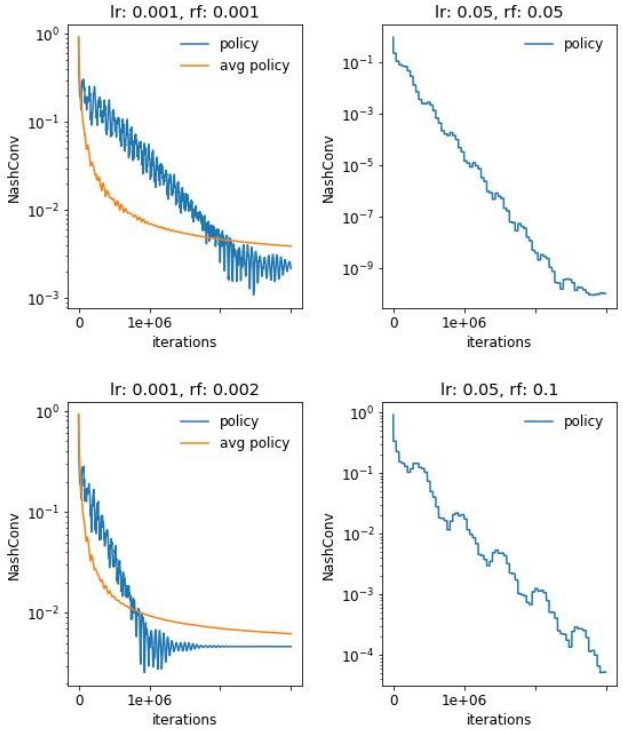
Regularization centered around [0.19, 0.72, 0.07].



Solution : [0.13 0.76 0.09]

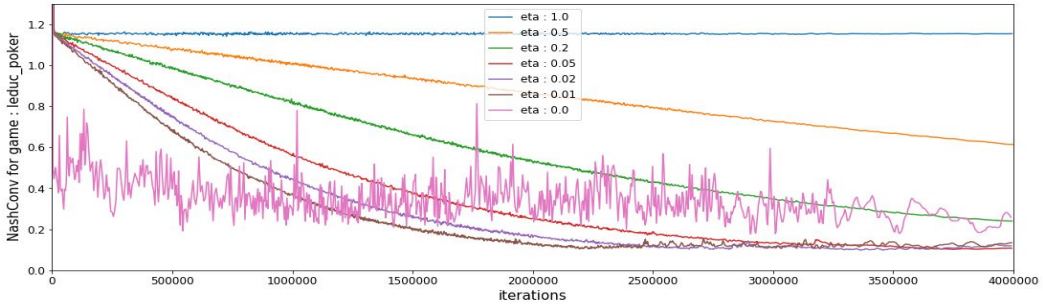
Friction FoReL: Experiments in Sequential Games

Convergence in Sequential Imperfect Information Games (Kuhn Tabular):

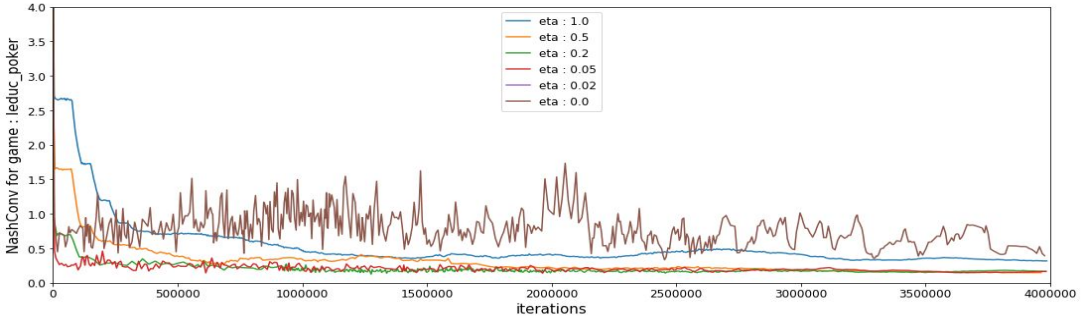


Convergence in Sequential Imperfect Information Games (Leduc with Neural Network and a NeuRD loss):

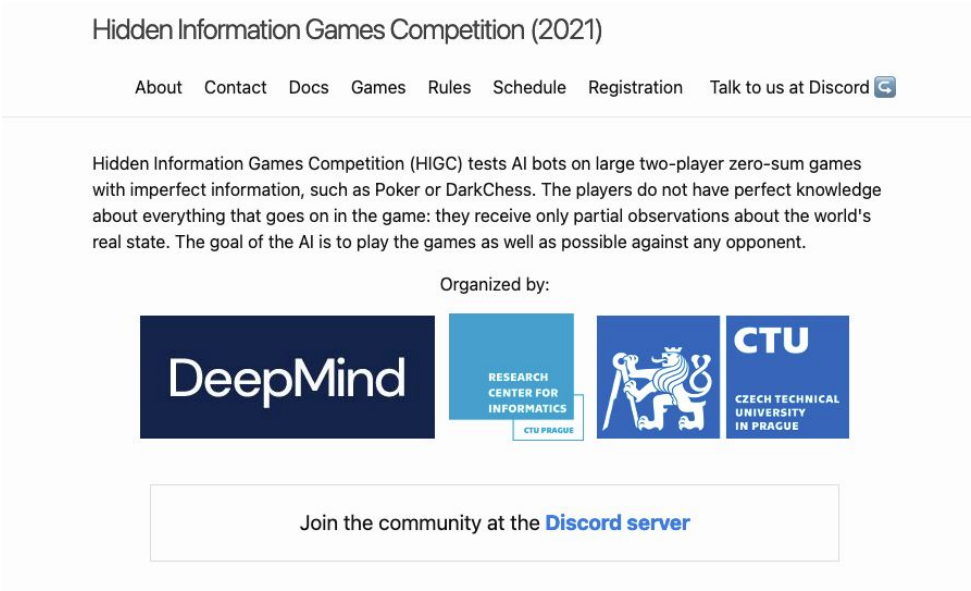
fixed mu and decaying eta (to a fixed eta):



fixed eta and refreshed mu



Hidden Information Game Competition (HIGC)



RECONNAISSANCE BLIND CHESS

- Reconnaissance Blind Chess
- Gin Rummy
- One hidden game!

higcompetition.info/

DeepMind

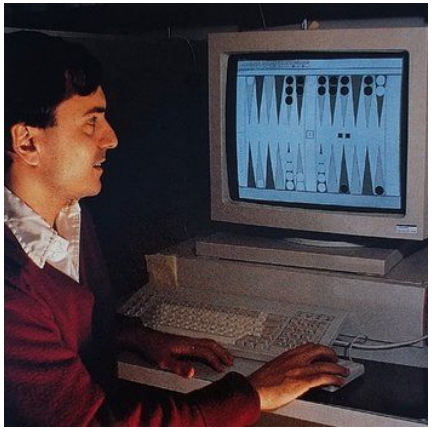
2 A Plea to "Go Wide": Beyond Zero-Sum and Domain-Specific Evaluation

Games, RL, and AI

Arthur Samuel - Checkers

IBM - Chess (DeepBlue)

Poker: DeepStack & Libratus



Gerald Tesauro - Backgammon

DeepMind - Go (AlphaGo)

Minimax

John von Neumann 1928

Max-min: P1 looks for a π_1 such that

$$v_1 = \max_{\pi_1} \min_{\pi_2} u_1(\pi_1, \pi_2)$$

Min-max: P1 looks for a π_1 such that

$$v_1 = \min_{\pi_2} \max_{\pi_1} u_1(\pi_1, \pi_2)$$

In two-player, zero-sum these are the same!

---> The Minimax Theorem

Consequences of Minimax

The optima $\pi^* = (\pi_1^*, \pi_2^*)$

- These exist! (They sometimes might be stochastic.)
- Called a minimax-optimal joint policy. Also, a Nash equilibrium.
- They are **interchangeable:**
- $\forall \pi^*, \pi^{*\prime} \Rightarrow (\pi_1^*, \pi_2^{*\prime}), (\pi_1^{*\prime}, \pi_2^*)$ also minimax-optimal
- Each policy is a best response to the other.

Minimax (Outside Two-Player Zero-Sum Games)

Max-min: Player i looks for a π_i such that:

$$v_i^{maxmin} = \max_{\pi_i} \min_{\pi_{-i}} u_i(\pi_i, \pi_{-i})$$

(Paranoid)

Minimax (Outside Two-Player Zero-Sum Games)

Max-min: Player i looks for a π_i such that:

$$v_i^{maxmin} = \max_{\pi_i} \min_{\pi_{-i}} u_i(\pi_i, \pi_{-i})$$

(Paranoid)

Min-max: Player j looks for a π_j (*against* π_i, i.e. in π_{-i}) such that

$$v_i^{minmax} = \min_{\pi_{-i}} \max_{\pi_i} u_i(\pi_i, \pi_{-i})$$

(Optimistic)

Nash Equilibrium

A joint policy used by all n players: $\pi = (\pi_1, \pi_2, \cdots, \pi_n)$

such that $\forall i, u_i(\pi) \geq \max_{\pi_i'} u_i(\pi_i', \pi_{-i})$

Properties of Nash Equilibria

Suppose each player computes an equilibrium: π^A, π^B, π^C

Properties of Nash Equilibria

Suppose each player computes an equilibrium: π^A, π^B, π^C

- $(\pi_1^A, \pi_2^B, \pi_3^C)$ is generally not an equilibrium

Properties of Nash Equilibria

Suppose each player computes an equilibrium: π^A, π^B, π^C

- $(\pi_1^A, \pi_2^B, \pi_3^C)$ is generally not an equilibrium
- Each equilibrium might have different values for all players

Properties of Nash Equilibria

Suppose each player computes an equilibrium: π^A, π^B, π^C

- $(\pi_1^A, \pi_2^B, \pi_3^C)$ is generally not an equilibrium
- Each equilibrium might have different values for all players
- Which equilibrium should you “choose”?
  - Equilibrium selection problem

Properties of Nash Equilibria

Suppose each player computes an equilibrium: π^A, π^B, π^C

- $(\pi_1^A, \pi_2^B, \pi_3^C)$ is generally not an equilibrium
- Each equilibrium might have different values for all players
- Which equilibrium should you “choose”?
  - Equilibrium selection problem
- Also PPAD-Hard (Daskalakis, Goldberg, Papadimitriou + Chen & Deng)

Some Thought Questions

- How do you tell if chess program is super-human?

Some Thought Questions

- How do you tell if chess program is super-human?
- What is “super-human” Iterated Prisoner’s Dilemma?

Some Thought Questions

- How do you tell if chess program is super-human?
- What is “super-human” Iterated Prisoner’s Dilemma?
- If minimax is optimal, why doesn’t it win RoShamBo competitions?

Some Thought Questions

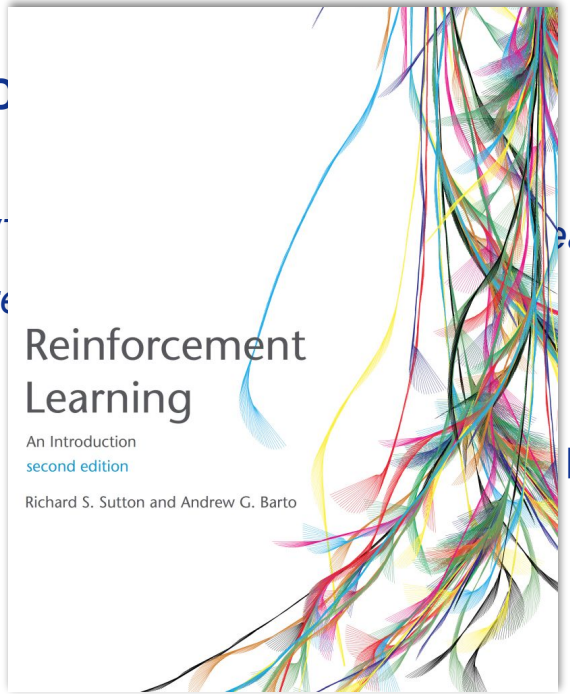
- How do you tell if chess program is super-human?
- What is “super-human” Iterated Prisoner’s Dilemma?
- If minimax is optimal, why doesn’t it win RoShamBo competitions?
- What should **general agents** learn in a multiagent setting?

Back to the Essentials: What's the Goal?

C

"

re



eaking, is to maximize the total amount of reward it

policy

π^*

Hindsight Rationality

$$\cdots \leftarrow \pi_1^{t-2} \leftarrow \pi_1^{t-1} \leftarrow \pi_1^t$$  Agent 1

$$\cdots \leftarrow \pi_2^{t-2} \leftarrow \pi_2^{t-1} \leftarrow \pi_2^t$$  Agent 2

$$\vdots \qquad \vdots$$

$$\cdots \leftarrow \pi_n^{t-2} \leftarrow \pi_n^{t-1} \leftarrow \pi_n^t$$  Agent N

Hindsight Rationality

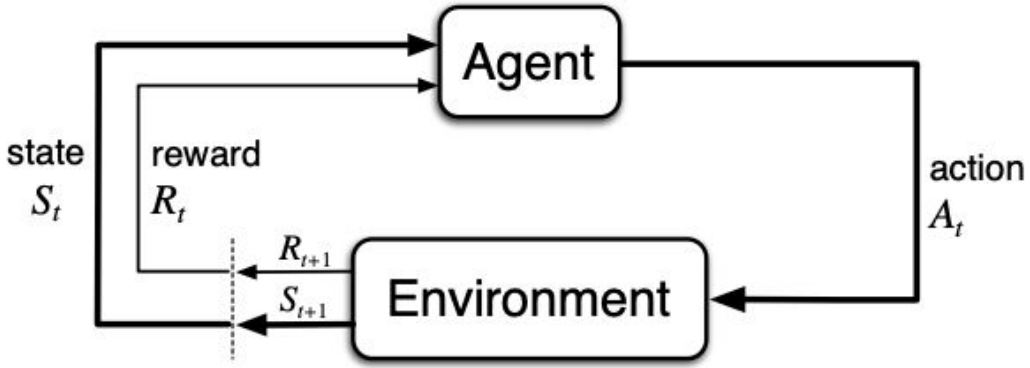
$$\cdots \leftarrow \pi_i^{t-2} \leftarrow \pi_i^{t-1} \leftarrow \pi_i^{t}$$

Agent i

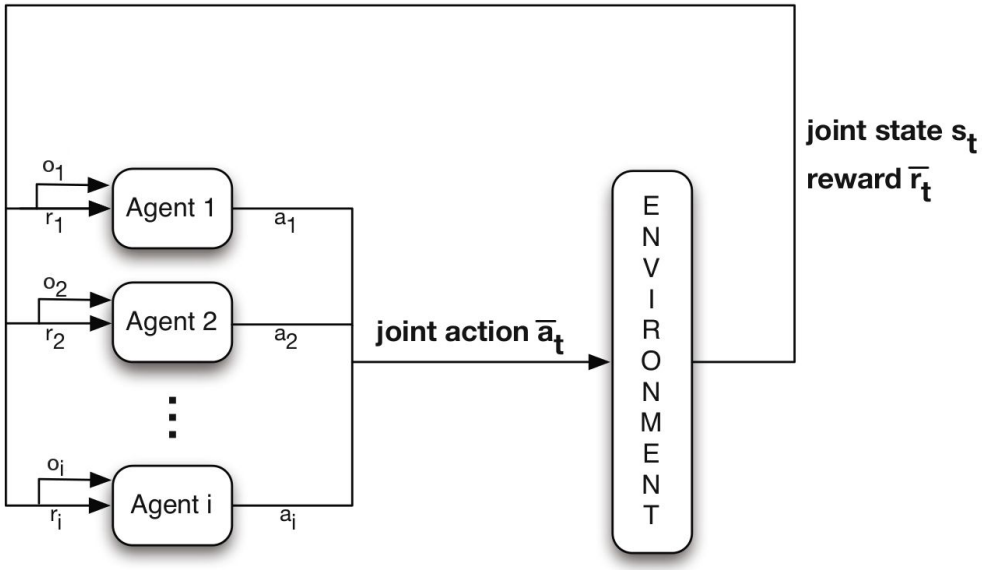
Hindsight Rationality

$$\cdots \leftarrow \pi_i^{t-2} \leftarrow \pi_i^{t-1} \leftarrow \pi_i^{t}$$

Agent i



Hindsight Rationality in Multiagent Environments



Hindsight Rationality in Multiagent Environments

Agent i perceives:

$$\cdots, (o_i^{t-2}, r_i^{t-2}), (o_i^{t-1}, r_i^{t-1}), (o_i^t, r_i^t)$$

- Set of **deviations** considered: Φ

Hindsight Rationality in Multiagent Environments

Agent i perceives:

$$\cdots, (o_i^{t-2}, r_i^{t-2}), (o_i^{t-1}, r_i^{t-1}), (o_i^t, r_i^t)$$

- Set of **deviations** considered: Φ
- π_i^t chosen in a way that minimizes regret w.r.t. Φ

Hindsight Rationality in Multiagent Environments

Agent i perceives:

$$\cdots, (o_i^{t-2}, r_i^{t-2}), (o_i^{t-1}, r_i^{t-1}), (o_i^t, r_i^t)$$

- Set of **deviations** considered: Φ
- π_i^t chosen in a way that minimizes regret w.r.t. Φ
- … based on history of **correlated play**!

Hindsight Rationality in Multiagent Environments

Agent i perceives:

$$\dots, (o_i^{t-2}, r_i^{t-2}), (o_i^{t-1}, r_i^{t-1}), (o_i^t, r_i^t)$$

- Set of **deviations** considered: Φ
- π_i^t chosen in a way that minimizes regret w.r.t. Φ
- … based on history of **correlated play**!

→ Classes of (extensive-form) correlated equilibria

Hindsight Rationality

Deviation sets Φ

- Functions of the entire sequence of past plays
- Sequential deviations too (not just entire policy deviations)
- New counterfactual deviations and associated equilibria in self-play

arXiv.org > cs > arXiv:2012.05874

Computer Science > Computer Science and Game Theory

[Submitted on 10 Dec 2020 (v1), last revised 17 Dec 2020 (this version, v2)]

Hindsight and Sequential Rationality of Correlated Play

Dustin Morrill, Ryan D'Orazio, Reca Sarfati, Marc Lanctot, James R. Wright, Amy Greenwald, Michael Bowling

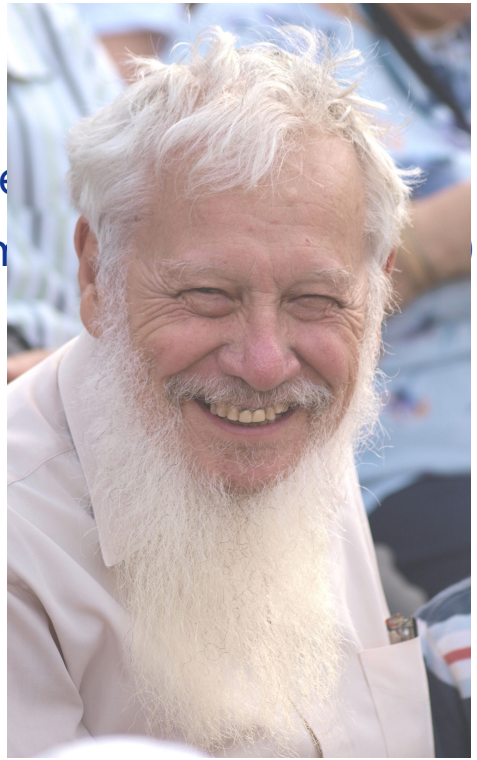
Efficient Deviation Types and Learning for Hindsight Rationality in Extensive-Form Games

Dustin Morrill, Ryan D'Orazio, Marc Lanctot, James R. Wright, Michael Bowling, Amy Greenwald

→ See also first COMARL seminar by Michael Bowling

Correlated Equilibrium (CE)

- CE arise as a *result* of learning
- (More) compatible with Bayesian perspective
- Compatible with a prescriptive agenda (via n
- Tractable!! (to find one)



Robert Aumann

Correlated Equilibrium Example

	Rock	Paper	Scissors
Rock	0, 0	0, 1	1, 0
Paper	1, 0	0, 0	0, 1
Scissors	0, 1	1, 0	0, 0

Figure 7.6: Shapley's Almost-Rock-Paper-Scissors game.

Shoham & Leyton-Brown '09

Correlated Equilibrium Example

	Rock	Paper	Scissors
Rock	0, 0	0, 1	1, 0
Paper	1, 0	0, 0	0, 1
Scissors	0, 1	1, 0	0, 0

Figure 7.6: Shapley's Almost-Rock-Paper-Scissors game.

Shoham & Leyton-Brown '09

Correlated Equilibrium Example

	Rock	Paper	Scissors
Rock	0, 0	0, 1	1, 0
Paper	1, 0	0, 0	0, 1
Scissors	0, 1	1, 0	0, 0

Figure 7.6: Shapley's Almost-Rock-Paper-Scissors game.

9. Then recommend each player their

r recommendation?

Shoham & Leyton-Brown '09

Correlated Equilibrium Example

	Rock	Paper	Scissors
Rock	0, 0	0, 1	1, 0
Paper	1, 0	0, 0	0, 1
Scissors	0, 1	1, 0	0, 0

Figure 7.6: Shapley's Almost-Rock-Paper-Scissors game.

9. Then recommend each player their

r recommendation?

ty ⅙. Is this a CE? If so, which one

Shoham & Leyton-Brown '09

Calibrated Learning [Foster & Vohra '97]

The following process converges to a CE:

Repeat over many trials t → T:

1. Compute a "calibrated forecast" of the opponents' policies (i.e. asymptotically consistent as t → +inf)
2. Best respond to the forecast

Claim: basis for a "best response stream" outside zero-sum

Bayesian Perspectives and Meta-Learning

Meta-learning of Sequential Strategies

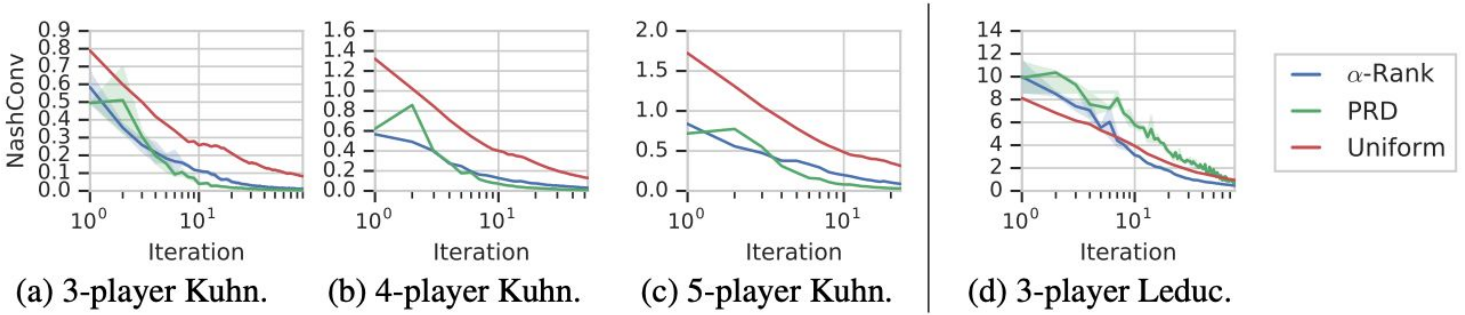
Pedro A. Ortega, Jane X. Wang, Mark Rowland, Tim Genewein, Zeb Kurth-Nelson, Razvan Pascanu, Nicolas Heess, Joel Veness, Alex Pritzel, Pablo Sprechmann, Siddhant M. Jayakumar, Tom McGrath, Kevin Miller, Mohammad Azar, Ian Osband, Neil Rabinowitz, András György, Silvia Chiappa, Simon Osindero, Yee Whye Teh, Hado van Hasselt, Nando de Freitas, Matthew Botvinick, and Shane Legg
DeepMind

Meta-trained agents implement Bayes-optimal agents

Vladimir Mikulik,* Grégoire Delétang,* Tom McGrath,* Tim Genewein,*
Miljan Martic, Shane Legg, Pedro A. Ortega†
DeepMind
London, UK

A Generalized Training Approach for MARL
[Muller et al. '19]

- Game-theoretic training:
  - (Meta-)solve empirical game
  - Find best response oracles
- Outside two-player zero-sum:
  - Use α-Rank as a tractable solution concept

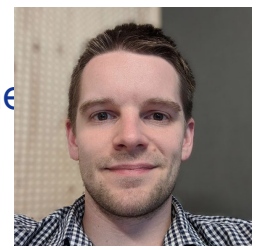


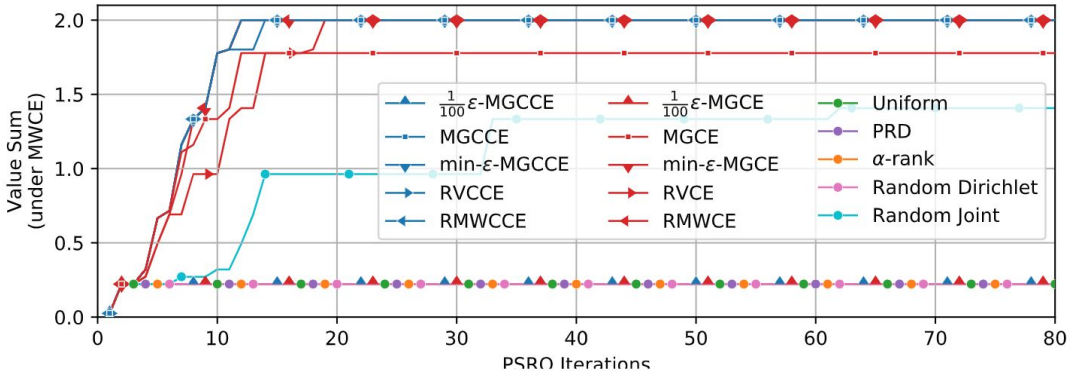
(a) 3-player Kuhn. (b) 4-player Kuhn. (c) 5-player Kuhn. (d) 3-player Leduc.

Figure 4: Results for poker domains with more than 2 players.

Joint Policy-Space Response Oracles (JPSRO)
[Marris et al. '21]

- Game-theoretic training:
  - (Meta-)solve empirical game
  - Find best response oracles
- Finds **joint** distribution
- New CE meta-solvers driving by **Gini impurity** (eq. selection)
- Converges to CE & CCE
- Works for n-player general-su
- Stochastic meta-solver poss



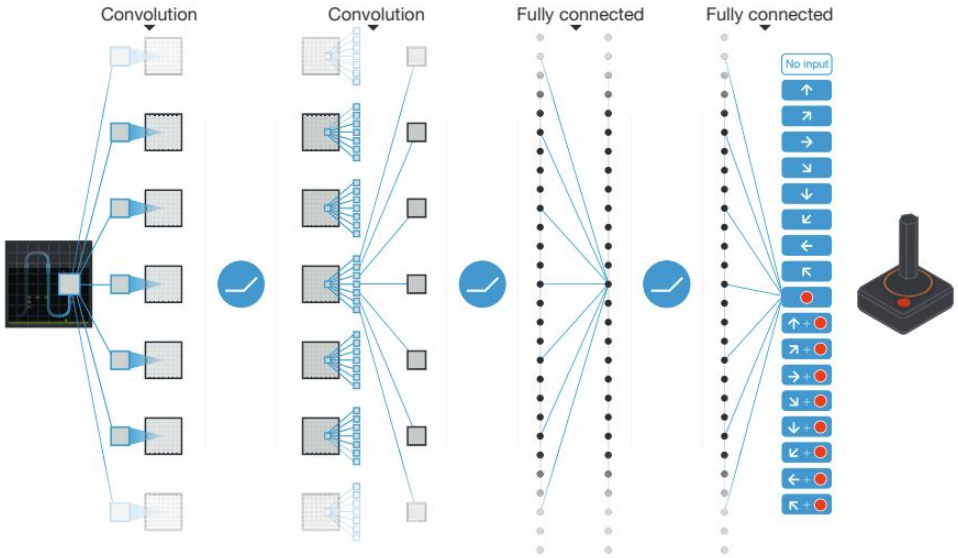
2-player Trade Comm (simplified trading game)

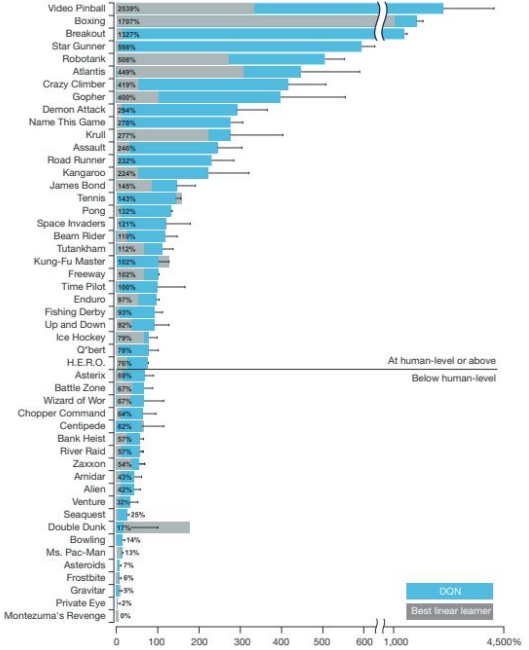
→ @ICML

A Plea to “Go Wide”: Beyond Domain-Specific Evaluation

A Plea to “Go Wide”: Beyond Domain-Specific Evaluation

- Progress in Deep RL benefited from the **generality** of Atari





A Plea to “Go Wide”: Beyond Domain-Specific Evaluation

- Progress in Deep RL benefited from the **generality** of Atari

BLOG POST
RESEARCH

31 MAR 2020

Agent57: Outperforming the human Atari benchmark

A Plea to "Go Wide": Beyond Domain-Specific Evaluation

- Progress in Deep RL benefited from the **generality** of Atari
- Games provide the formalism, have been the heart of multiagent RL

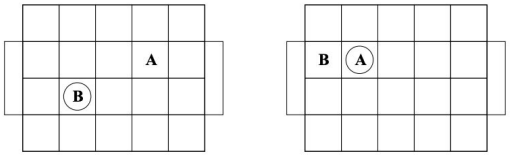
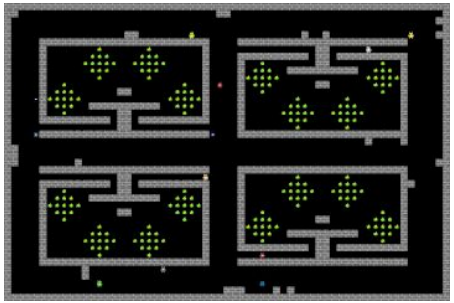


Figure 2: An initial board (left) and a situation requiring a probabilistic choice for A (right).



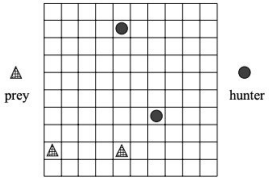
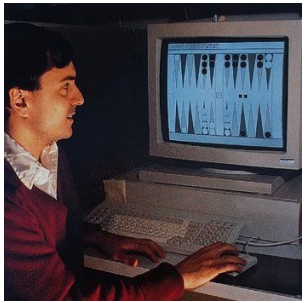


Figure 1: A 10 by 10 grid world.



A Plea to “Go Wide”: Beyond Domain-Specific Evaluation

- Progress in Deep RL benefited from the **generality** of Atari
- Games provide the formalism, have been the heart of multiagent RL
- The MARL community is split across many different types:
  - Competitive vs. Cooperative vs. Mixed / Social Dilemmas

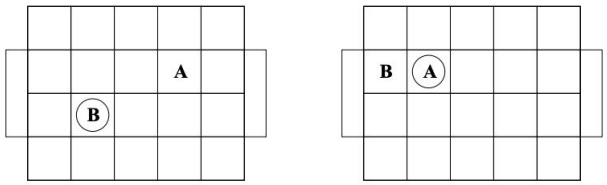
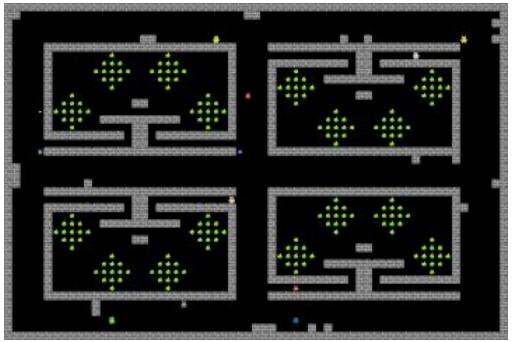


Figure 2: An initial board (left) and a situation requiring a probabilistic choice for A (right).



A Plea to "Go Wide": Beyond Domain-Specific Evaluation

- Progress in Deep RL benefited from the **generality** of Atari
- Games provide the formalism, have been the heart of multiagent RL
- The MARL community is split across many different types:
  - Competitive vs. Cooperative vs. Mixed / Social Dilemmas
- A generally intelligent agent is capable **across many environments**!

$$\Upsilon(\pi) := \sum_{\mu \in E} 2^{-K(\mu)} V_{\mu}^{\pi}.$$

Measure of Intelligence

Sum over environments

Complexity penalty

Value achieved

"Intelligence measures an agent's ability to achieve goals in a wide range of environments"

-- Shane Legg & Marcus Hutter '07, "Universal Intelligence"

A Plea to “Go Wide”: Beyond Domain-Specific Evaluation

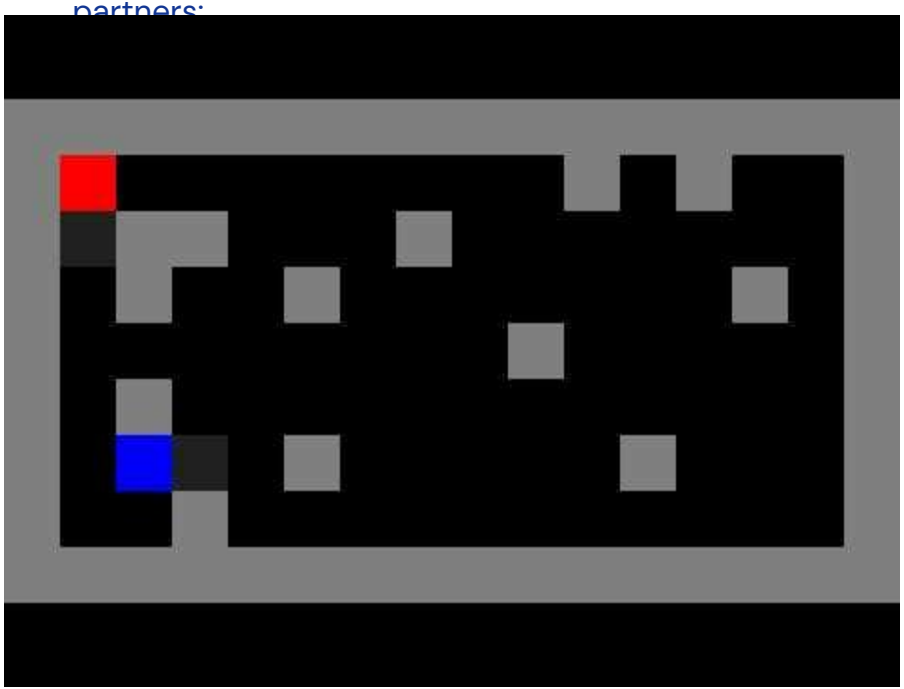
- Progress in Deep RL benefited from the **generality** of Atari
- Games provide the formalism, have been the heart of multiagent RL
- The MARL community is split across many different types:
  - Competitive vs. Cooperative vs. Mixed / Social Dilemmas
- A generally intelligent agent is capable **across many environments**!
- An artificial general intelligence (AGI) can handle all cases

A Plea to "Go Wide": Beyond Domain-Specific Evaluation

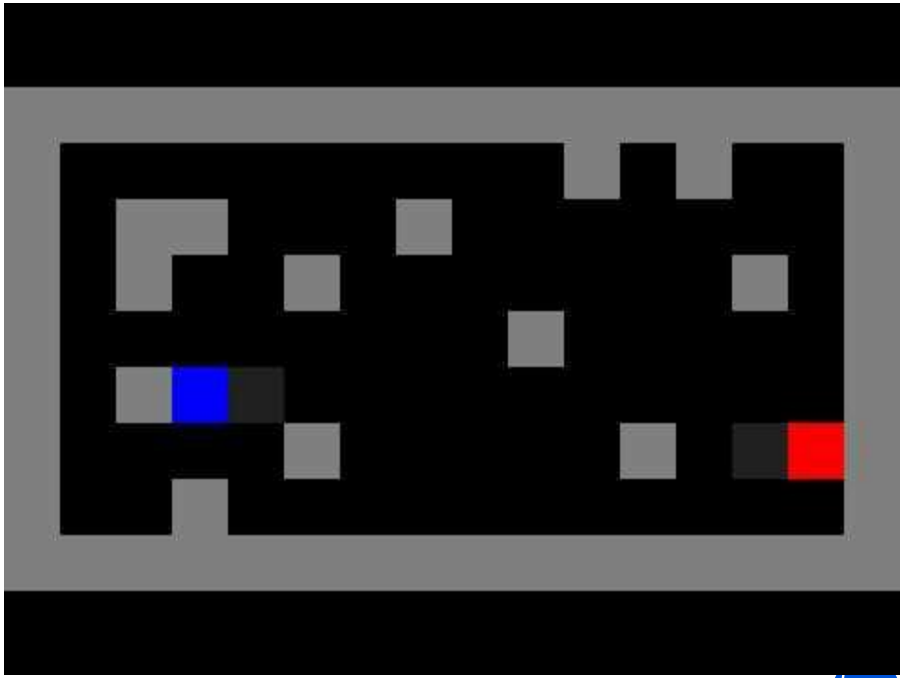
- Progress in Deep RL benefited from the **generality** of Atari
- Games provide the formalism, have been the heart of multiagent RL
- The MARL community is split across many different types:
  - Competitive vs. Cooperative vs. Mixed / Social Dilemmas
- A generally intelligent agent is capable **across many environments**!
- An artificial general intelligence (AGI) can handle all cases
- Games:
  - Are externally defined or inspired by human problems
  - Have been traditional benchmarks of rationality for thousands of years
  - Have formally-defined interactions
  - Can be easily simulated and run many times
  - Are enjoyable to play and easy to demonstrate with humans

Why Generality in MARL?

Independent learners with their training partners:

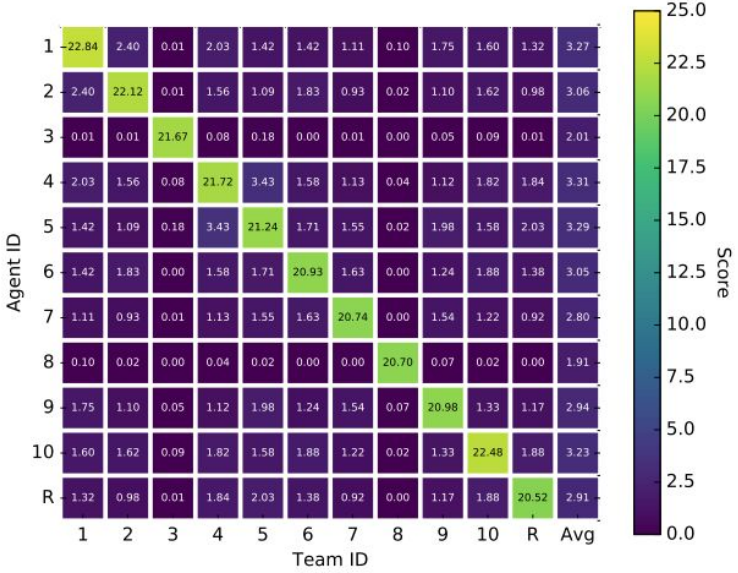


Independent learners with similarly-trained partners:

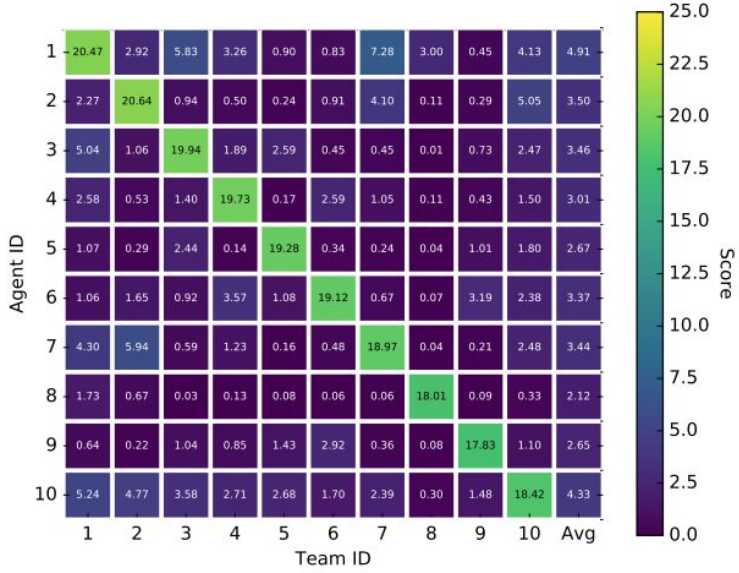


Self-Play / Independent Agents Do Not Generalize

The Hanabi Challenge: A New Frontier for AI Research [Bard et al. '19]



(a) Ad-hoc results for two players.



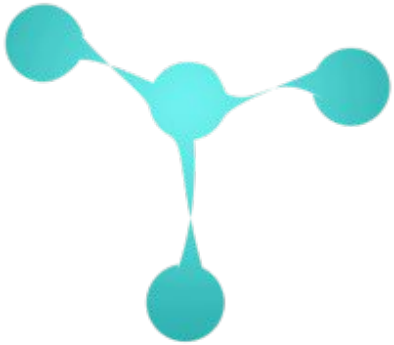
(b) Ad-hoc results for four players.

How Do We Get There?

Some ideas:

1. More domains in empirical evaluations
2. More openly available code … *for MARL specifically*
3. Increased focus on *ad-hoc setting* for evaluation
    a. Especially with human(s) in the loop, where possible
4. *Wider* and more dynamic evaluation regimes

Importance of Open-Source and Widely Usable Libraries



OpenSpiel: A Framework for RL in Games

github.com/deepmind/open_spiel/

Supports:

- n-player games
- Zero-sum, coop, general-sum
- Perfect / imperfect info
- Simultaneous-move games

- One main API for all games
- Atari Learning Env. of multiagent / games
- Release 1.0 coming in August (!)
  - with new mean-field game API

MAVA: A New Open-Source Framework for MARL

- Scalable multiagent training framework:
  - Built on open-source technologies:
    - Reverb
    - Launchpad
    - Acme
- Integration with **many environments / libraries**!
  - PyMARL
  - OpenSpiel
  - PettingZoo
  - Flatland
  - Robocup
  - Starcraft Multi-Agent Challenge (SMAC)

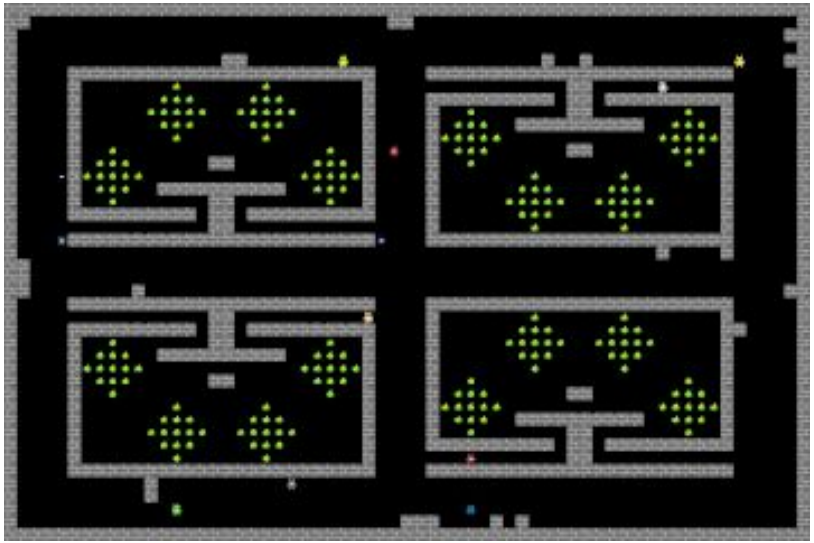
[Pret
Tessera, S

https://github.com/instadeepai/Mava

Scalable Evaluation of MARL with Melting Pot

[Leibo, Duéñez-Guzmán, Vezhnevets, Agapiou, et al. '21]

- Compare algorithms/agents
- Focus on **generalization**
- Over 80 unique test scenarios!
- Eval in held out test *scenarios*
- Agnostic to training method



Potential General Evaluation Regimes

- A suite of general learning agents that can play all sides/roles
- A suite of games (domains) to evaluate on

Agent match-ups:

- Agent versus fixed reference set
- Agent versus agent

- Agent sampling distr: fixed versus adaptive
- Inter-match memory: none (blank slate) vs. learning (lifelong)

Potential General Evaluation Regimes: A vs. fixed ref set

à la Melting Pot. Evaluating agent A:

... with test-time adaptation

For all games in selected games, G:

- Decide on a fixed reference set Ref(G)
  - E.g. self-play RL benchmark, known strategies, etc.
- Evaluate A over/with other agents in Ref(G)
- Get overall return V(G)

Report V(G) for all selected games

Potential General Evaluation Regimes: A vs. fixed ref set

à la Melting Pot. Evaluating agent A:

... with test-time adaptation

Agent A is always learning.

For all games in selected games, G:

- Ensure observations encode identification of G.
- Decide on a fixed reference set Ref(G)
  - E.g. self-play RL benchmark, known strategies, etc.
- Evaluate A over/with other agents in Ref(G)
- Get overall return V(G)

Report V(G) for all selected games

Potential General Evaluation Regimes: A vs. fixed ref set

à la Melting Pot. Evaluating agent A:

… with adaptive distr.

For all games in selected games, G:

- Decide on a fixed reference set Ref(G)
  - E.g. self-play RL benchmark, known strategies, etc.
- Initialize OppDist(Ref(G)) to uniform
- For epochs t = 1 … T:
  - Evaluate A over/with other agents in Ref(G) using OppDist
  - Get overall return V(G, t)
  - Adjust OppDist adversarially

Report V(G, t = 1 .. T) for all selected games.

Potential General Evaluation Regimes: Agent vs. Agent

- Similarly to fixed reference set, except:
  - Reference set is not fixed
  - Other agents might be in reference sets

Goal: fully online, continuous, lifelong evaluation across many environments

Conclusions & Summary

- Game-theoretic approaches to partially observable games
  - Inspired by two-player zero-sum games
  - Shown to scale to very large domains
  - Two main streams: best response and no-regret
  - Many ideas can be generalized outside two-player zero-sum
  - Correlated equilibria as link between prescriptive/descriptive views

Conclusions & Summary

- Game-theoretic approaches to partially observable games
  - Inspired by two-player zero-sum games
  - Shown to scale to very large domains
  - Two main streams: best response and no-regret
  - Many ideas can be generalized outside two-player zero-sum
  - Correlated equilibria as link between prescriptive/descriptive views
- Let's "Go Wide": evaluate agents across many environments!
  - Pursuit of general agents requires general evaluation
  - Many efforts to help make this happen
  - "Never been a better time than right now" :)

-- RHCP, *Give it Away*

Thank You! ... Any Questions?

Marc Lanctot

lanctot@deepmind.com

mlanctot.info/