Fictitious Cross-Play: Learning Global Nash Equilibria in Mixed Cooperative-Competitive Games

Zelai Xu, Yancheng Liang, Chao Yu, Yu Wang, Yi Wu. AAMAS 2023

Yancheng Liang

2023-01-18

Background

- In a mixed cooperative-competitive game, two teams of agents play against each other.

Example: Google-research football.
The 11-vs-11 full version is extremely challenging and serves as an important benchmark for learning large-scale games.

- The game is two-player (team) zero-sum for both teams, and it is cooperative for members in the same team
- We proposes fictitious cross-play (FXP) to learn “better” policies
- It defeats SOTA models in Google-research football

Problem Setup

- We have two N-player team, with joint policy

$$\pi_{joint} = \prod_{i=1}^{2} \pi_{t_i} = \prod_{i=1}^{2} \prod_{j=1}^{N} \pi_{ij}$$

- The utility (reward) function is both cooperative

$$U_{i,1}(\pi) = U_{i,2}(\pi) = \cdots = U_{i,N}(\pi) = U_{t_i}(\pi)$$

- and competitive (zero-sum with respect to the team)

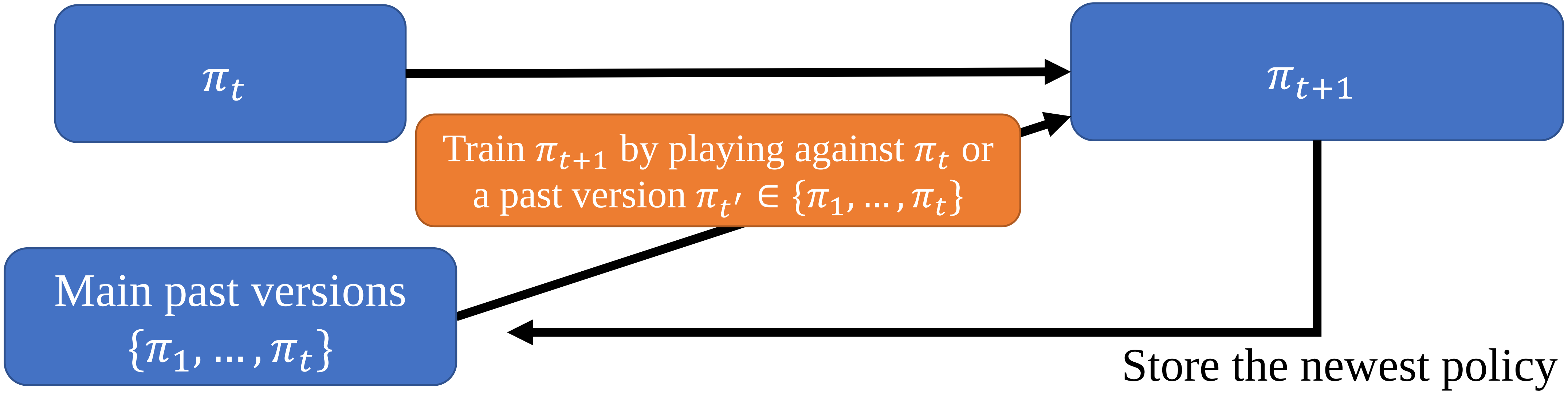
$$U_{t_1}(\pi) + U_{t_2}(\pi) = 0$$

- Therefore, we can define a local (individual) NE if $\pi_k = BR(\pi_{-k}), \forall k$
- And a global (team) NE if $\pi_{t_i} = BR\left(\pi_{t_{-i}}\right), \forall i \in \{1,2\}$

Self-Play

- Vanilla self-play

- Fictitious replay



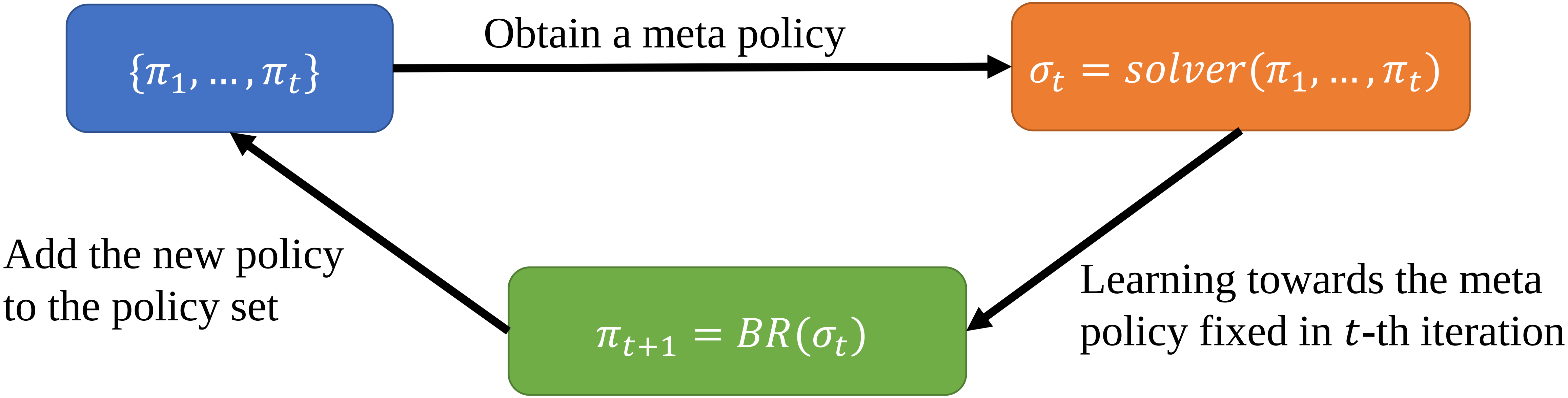
Self-Play

- Self-play is the most popular paradigm for multi-agent reinforcement learning

Starcraft II	DoTA 2	Quake III Arena (FPS)
Two-player zero-sum	Mixed cooperative-competitive	Mixed cooperative-competitive
Population-based training (PBT)	Fictitious replay	Population-based training (PBT)

- Self-play agents often perform terribly when the opponent's policy are "out-of-distribution".
- All these methods maintain a "population" of policies to improve policy diversity.

Policy-Space Response Oracles (PSRO)



- PSRO is different from self-play on the learning problem, while
  - Self-play learns to both cooperate with teammates and compete with varying opponents
  - PSRO only learns a cooperative game, since the opponent policy is fixed (part of the stationary environment)

A Motivating Example Game

- For two-player zero-sum games, many SP-based algorithms are guaranteed to converge to NE, but they fail to solve the following mixed cooperative-competitive game due to partial observability

Example Game

$$U(0_N, 1_N) = C$$

$$U(0_N, y) = \epsilon N \sum_{i=1}^{N} y_i, \forall y \neq 1_N$$

$$U(x, y) = N \sum_{i=1}^{N} x_i - y_i, \forall x, y \neq 0_N$$

Theorem 4.1. Any "common self-play algorithm" will not converge to global NE if

$$\forall 1 \leq i \leq N, \pi_{-i}^{0}(0_N) \leq \frac{1}{N + 1 + 2C + \epsilon},$$

which has a probability of $1 - \frac{1}{e^{O(N)}}$

What is *Any "common self-play algorithm"*

- (Preference Preservation). We say a learning process is preference preservation if the relative ratio of choosing action x and y keeps increasing when all the past observed Q-function of x is larger than y, and the ratio updating rules are monotone with Q. To be more specific

$$\forall t' <= t, Q_i^{t'}(x) \geq Q_i^{t'}(y) \Rightarrow \frac{\pi_i^{t+1}(x)}{\pi_i^{t+1}(y)} \geq \frac{\pi_i^{t}(x)}{\pi_i^{t}(y)}$$

and

$$\forall t' \leq t, i, x, y, \ \frac{\pi_i^{t+1}(x)}{\pi_i^{t+1}(y)} = f_{i,x,y}^{t}\left(\{Q_x^s - Q_y^s\}_{s=0}^{t}, \{Q^s\}_{s=0}^{t}\right)$$

$$s.t.\ \nabla_{Q_x^{t'} - Q_y^{t'}} f_{i,x,y}^{t} \geq 0.$$

- This property holds for many SP-based algorithms, including FSP, Follow the Regularised Leader , Replicator Dynamics, Multiplicative Weights Update, Counter Factual Regret Minimization

How are PSRO and FXP (Ours) Different

- We show PSRO has better convergence property by training against fixed opponents

Theorem 4.2. For the same learning algorithm, in the example game, self play has a strictly smaller good initialization set $S_{SP} \subseteq S_{\mu} (S_{SP} \neq S_{\mu})$ *compared with training against fixed opponents* $\mu \in \{0_N, 1_N\}$.

- It also inspires our FXP to build a learning framework where the opponent's policy is relatively stationary

Self-play v.s. PSRO

Self-play

- Self-play is very efficient, since it always trains the strongest policy and the opponent is also the strongest version
- Theoretical guarantees on two-player zero-sum games no longer applicable for mixed cooperative-competitive games

PSRO

- Performs better by training against fixed opponents
- Needs to train a set of population, while it promotes diversity, it is very inefficient

Fictitious Cross-Play (FXP)

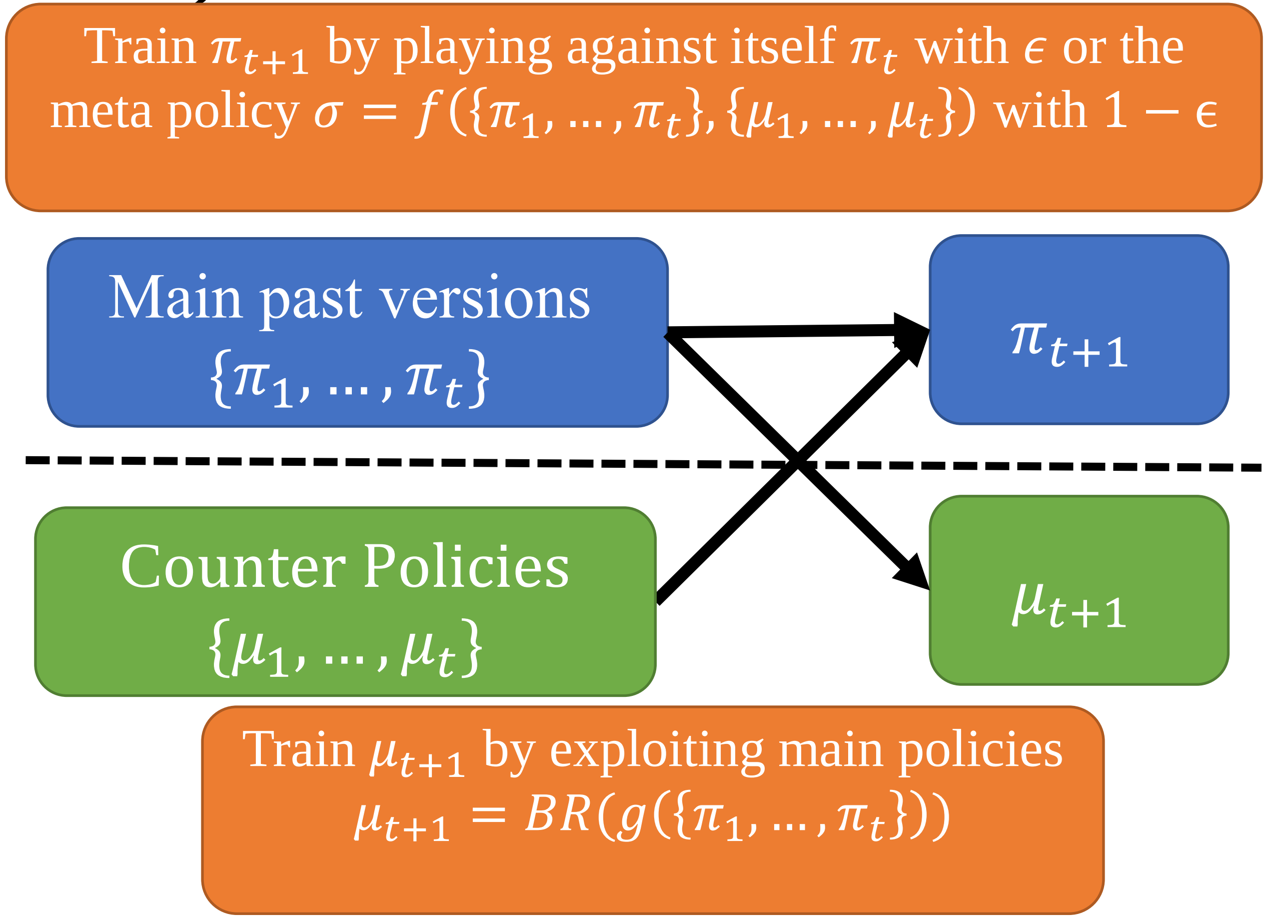
Algorithm 3: Fictitious Cross-Play (FXP)

Input: Initial main population and counter population with random policy $\Pi_M^1 = \{\pi_M^1\}, \Pi_C^1 = \{\pi_C^1\}$

for $t = 1, 2, \cdots, T$ **do**

- Update $U_{M+C}, U_{M\times C}$ by game simulations
- $\sigma_{M+C} \leftarrow$ meta-solver$_M(U_{M+C})$
- $\sigma_M, \sigma_C \leftarrow$ meta-solver$_C(U_{M\times C})$
- **for** *many episodes* **do**
  - Update π_M^{t+1} toward BR$(\eta\pi_M^{t+1} + (1-\eta)\sigma_{M+C}\Pi_{M+C}^t)$
  - Update π_C^{t+1} toward BR$(\sigma_M \Pi_M^t)$
- $\Pi_M^{t+1} \leftarrow \Pi_M^t \cup \{\pi_M^{t+1}\}$
- $\Pi_C^{t+1} \leftarrow \Pi_C^t \cup \{\pi_C^{t+1}\}$

Output: Population Π_M^{T+1}, Π_C^{T+1} and meta-policy σ_{M+C}

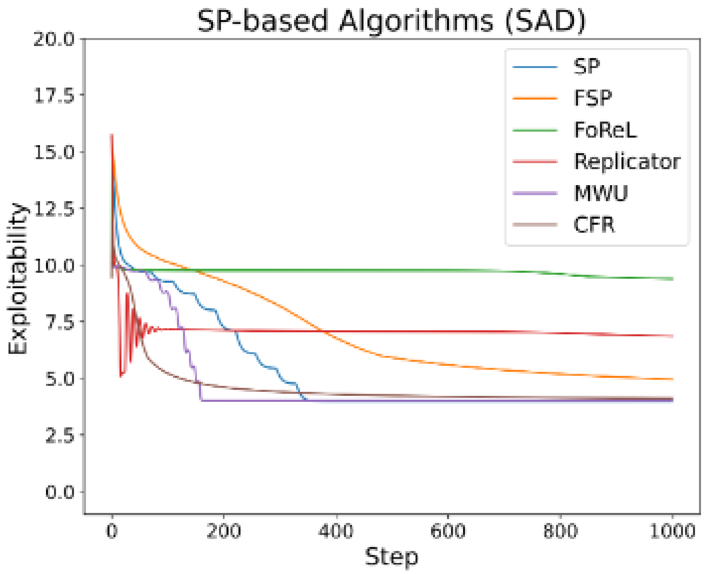


- FXP is an asymmetric framework
- One side (main policy) is SP that continuously improves it self
- The other side (counter policy) is PSRO training against main policies to find the weakness of them

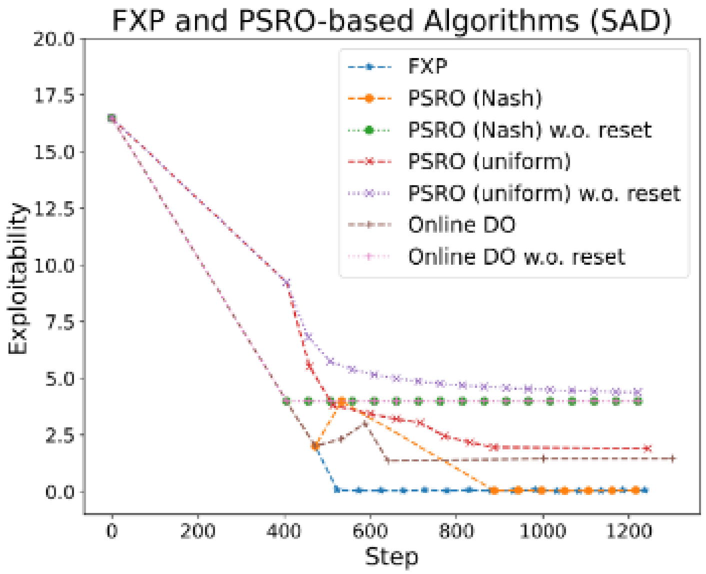
Illustrative environments

- Seek-attack-defend (SAD) game

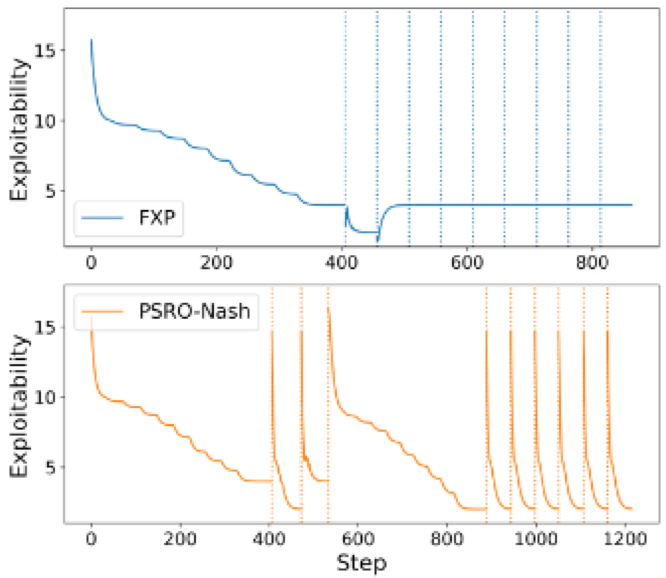
- Seek: every (seeking) player seek for $a_i \in \{x, x+1\}, (0 \leq x \leq M)$ and receives a total reward of $\sum a_i$. If team members do not seek cooperatively, they lose all reward.
  - The reward function is carefully designed so the agents need to learn to gradually simultaneously increase their seeking action a_i to avoid the failure of non-cooperative behaviors.
- Attack: at least two players attack the other team to make them lose all reward
- Defend: any player's defense protect the reward of its team



(a) None of SP-based algorithms converge to the global NE that has zero exploitability.



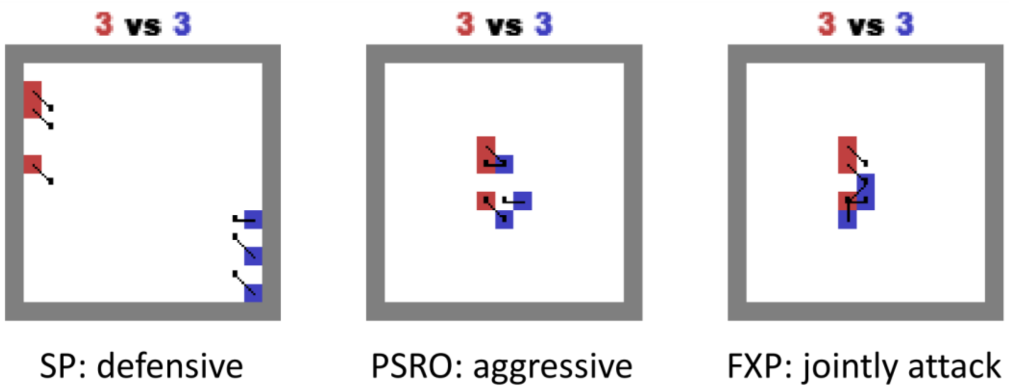
(b) Exploitability is computed on the meta policy. In *SAD* games FXP uses NE meta-solver.



(c) A vertical line means a new iteration.

- We plot the curves of exploitability (global NE has zero exploitability)
- For SP-based algorithms, none of them converge to global NE; some of them, including FP, FSP, CFR, MWU, converge to the local NE
- FXP and PSRO-Nash are the only two algorithms converge to the global NE; the reason behind that is FXP can leverage the skills learned before, while PSRO repeatedly learned some challenging skills, as shown in Figure (c)

Visualization: MAgent Battle



- Self-play: local NE policy that defends opponents at the corner. Self-play policies can be defeated by rushing to one agent and cooperative attack it
- PSRO: aggressively attack opponents because in each iteration, it learns to exploit a fixed opponent policy. It does not learn the best policy because the policy space is too large
- FXP: keep a safe distance and defend the opponents, but sometimes it will jointly attack one opponent if it finds the opponent to be defensive

Evaluation on Large-scale Games

- We replace the meta-solver with prioritized sampling
  - For main policy π_M's opponent π, we sample it proportional to $P(\pi\ wins\ \pi_M)$
  - For counter policy π_C's opponent π, we want them be comparable to accelerate the training of π_C (similar to curriculum learning), i.e., proportional to $P(\pi\ wins\ \pi_C)P(\pi_C\ wins\ \pi)$
- Performs well on Google research football 11-vs-11

Figure 5: Goal differences of FXP and other models against built-in AI of different levels.

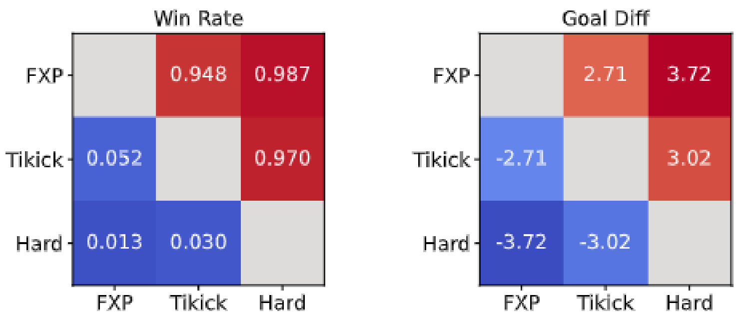


Figure 6: Head-to-head win rate evaluation between FXP, Tikick and built-in hard AI in 11-vs-11 full game.

Reference

- Oriol Vinyals et al. 2019, Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 575, 7782 (2019)
- Christopher Berner et al, 2019. Dota 2 with large scale deep reinforcement learning. arXiv preprint arXiv:1912.06680 (2019)
- Max Jaderberg et al, 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. Science 364, 6443 (2019), 859–865.