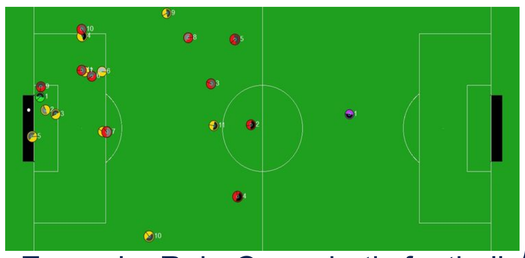
Team Selection in Multiagent Adversarial Team Games

Pranav Rajbhandari, Prithviraj Dasgupta, Donald Sofge
Information Technology Division, Naval Research Laboratory, Washington, D.C., United States

Multiagent Adversarial Team Games

- Environment with multiple teams of agents, each with a shared team goal
- ***Outcome*** at game's end describes which teams won
- Can be formalized as a ***Markov Decision Process (MDP)*** for each agent
- Has both cooperative and adversarial interactions



Example: RoboCup robotic football

Research Goals

- Train individual agent policies in multiagent adversarial team games
- Select optimal teams from a set of trained agents
  - Team members must cooperate
  - Teams must perform well against various opponents
  - Difficult task, as for n agents there are $\Theta(n^k)$ sized k teams

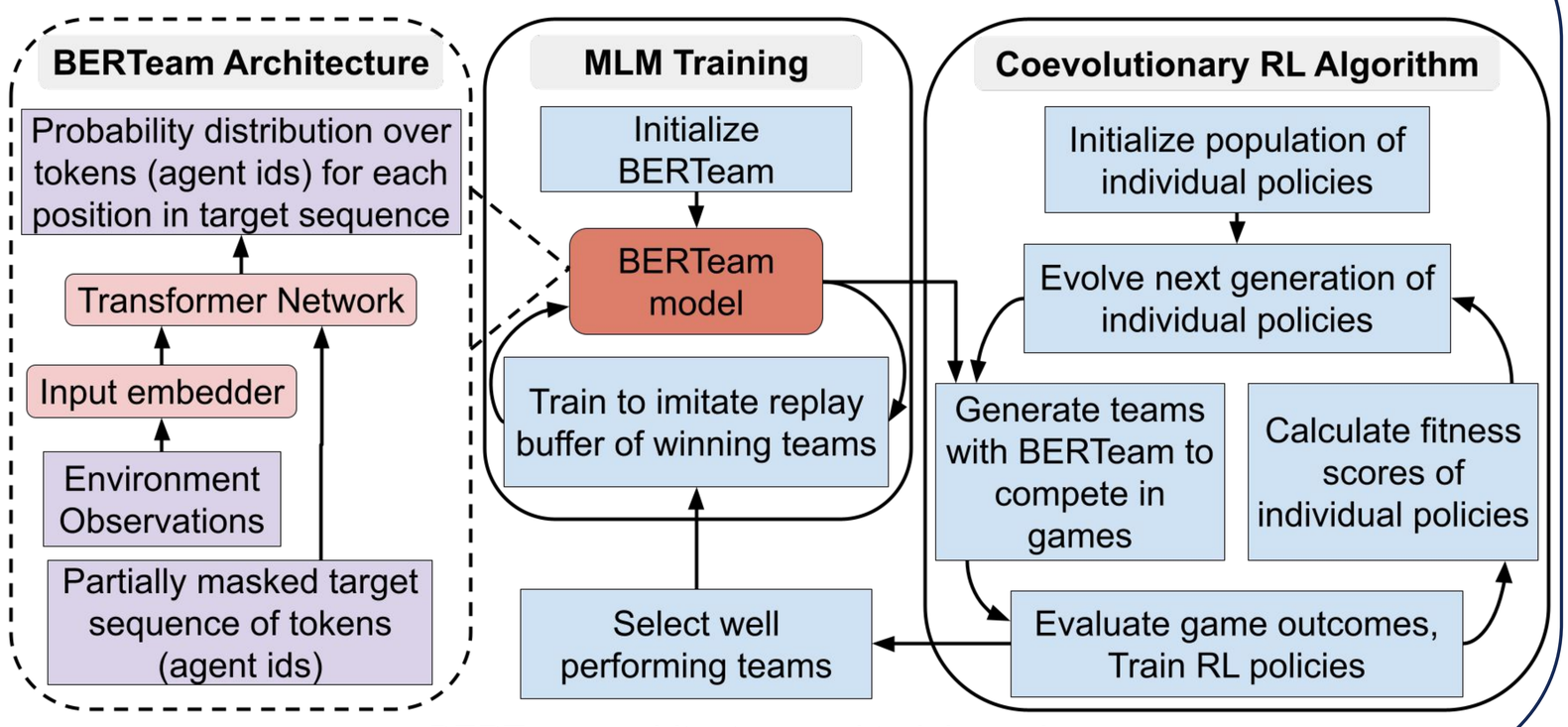
BERTeam

Architecture/Training

- Selecting a team from a set of fixed-policy agents is equivalent to generating an 'agent sequence'
- We approach this with ***BERTeam***, a transformer model trained through ***Masked Language Modeling (MLM)***
  - Predict masked tokens in a sequence from context
- BERTeam is trained to imitate dataset of winning teams

Training alongside Coevolution

- Update individual policies alongside training BERTeam (instead of using fixed-policy agents)
  - BERTeam obtains stronger agents to select from
  - Agents gain experience being on stronger teams/playing against stronger opponents
- ***Coevolutionary Reinforcement Learning*** improves a population of agents through ***Reinforcement Learning (RL)***, and replaces agents that perform poorly



BERTeam architecture and training scheme

Experiments

We evaluate the performance of BERTeam experimentally in 2v2 games of ***Pyquaticus***

Fixed Policies

- Trained BERTeam to generate teams from 7 fixed-policy agents
  - Offensive/Defensive agents of varying skill level
  - Randomly acting agent
- Calculated the team ***Elo*** of all possible teams to compare with BERTeam's distribution

BERTeam + Coevolution

- Trained BERTeam alongside Coevolutionary RL algorithm
  - Population: 50 ***PPO*** agents
- Estimated performance against fixed-policy agents
- Characterized trained policies as aggressive/defensive for analysis

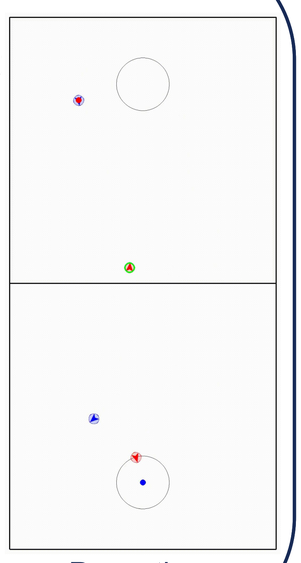
Comparison with MCAA

- ***Multiagent Coevolution for Asymmetric Agents (MCAA)***
- MCAA also decouples policy optimization (***MAP-Elites***) from its team selection, so we compare 4 hybrid algorithms
- Trained hybrid algorithms, evaluated games between teams selected from each

Pyquaticus

- Simulation of ***Aquaticus***, a robot Marine Capture-the-Flag game
- k-v-k game, each team must capture a flag from opponent's base without getting tagged

Aquaticus:
Robotic Marine Capture the Flag



Pyquaticus Screenshot

Results

Fixed Policies

- BERTeam chose the correct best team
  - {2,5}: One aggressive, one defensive agent
- BERTeam's top 7 similar to true ranking from Elos

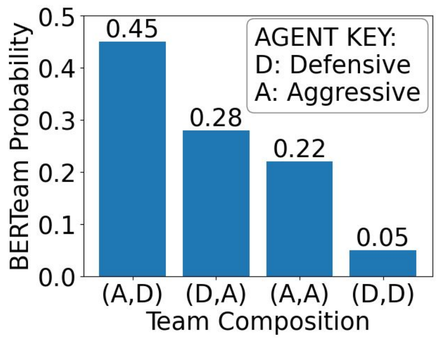
Team	True Rank/Elo		BERTeam Rank/Occurrence	
$\{2,5\}$	*1*	1388	*1*	0.14
$\{2,2\}$	*2*	1337	*2*	0.13
$\{2,3\}$	*3*	1135	*7*	0.06
$\{1,2\}$	*4*	1112	*4*	0.10
$\{0,2\}$	*5*	1097	*3*	0.10
$\{2,4\}$	*6*	1087	*5*	0.10
$\{2,6\}$	*7*	1035	*6*	0.07
$\{0,5\}$	*8*	975	*13*	0.03

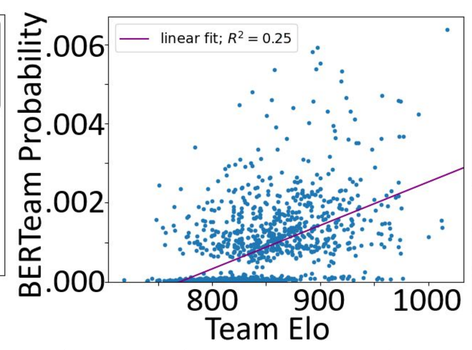
AGENT KEY:
Aggressive: Easy - 0, Medium - 1, Hard - 2
Defensive: Easy - 3, Medium - 4, Hard - 5
Random - 6

Comparison of true and predicted ranks

BERTeam + Coevolution

- Learned team composition is similar to best team with fixed-policy agent
  - One aggressive agent, one defensive agent
- BERTeam output correlates with team performance ($R^2 \approx 0.25$)
- Strongest team (≈1017 Elo) outperforms all fixed-policy teams that don't contain agent 2





BERTeam learned distribution on trained agents

Comparison with MCAA

- BERTeam outperforms MCAA independent of policy optimizer used
  - Compare algorithm Elo (**A**[Elo]) based on expected performance of a team selected by each algorithm
- BERTeam's network update takes significantly more time than MCAA's update

Policy Optimizer	Team Selection	𝔸[Elo]	Avg. update time of	
			Agents	Team Dist.
Coevolution	BERTeam	919	13 s/epoch	46 s/update
Coevolution	MCAA	817	13 s/epoch	≈ 0 s/update
MAP-Elites	BERTeam	883	36 s/epoch	45 s/update
MAP-Elites	MCAA	809	35 s/epoch	≈ 0 s/update

Relative performance of hybrid algorithms

Conclusions

- BERTeam is able to optimize a distribution of teams from fixed-policy agents
- BERTeam trained with coevolutionary RL is competitive against strong opponents not in training data
  - This algorithm outperforms MCAA in the setting of Pyquaticus

Future Work

- Team selection may be phrased as a ***normal form game*** between *coaches*
  - Update BERTeam algorithm to approach a ***Nash Equilibrium*** in this game
- Generate teams given partial information about environment/opponents
- Test on games with larger team sizes
  - Transformers can comfortably generate size ≈512 sequences