RLChina Reinforcement Learning Summer School

RLChina 2022

Opportunities and Challenges in Applying Multi-Agent Reinforcement Learning

Prof. FANG Fei

Leonardo Assistant Professor
School of Computer Science
Carnegie Mellon University

August 25, 2022

Machine Learning + Game Theory for Societal Challenges

Security & Safety

Zero Hunger

Environmental Sustainability

Transportation

Artificial Intelligence

Computational Game Theory

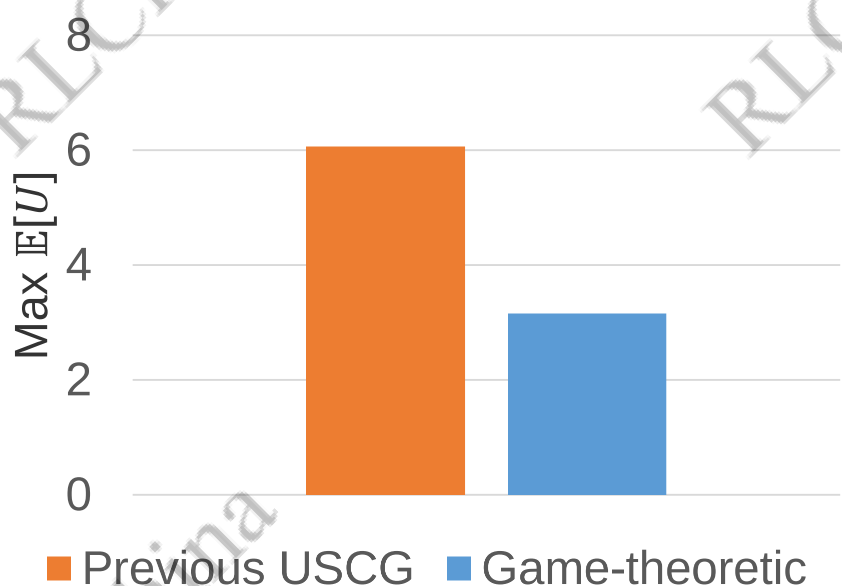
Machine Learning

Societal Challenges

Protect Ferry Line from Potential Attacks

- Defender-attacker security game
- Randomized patrol strategy
- Minimize attacker's maximum expected utility
- Solve through linear programming

Reduce potential risk by 50%



Deployed by US Coast Guard

Optimal Patrol Strategy for Protecting Moving Targets with Multiple Mobile Resources. Fei Fang, Albert Xin Jiang, Milind Tambe. In AAMAS-13

In collaboration with US Coast Guard

Protect Wildlife from Poaching

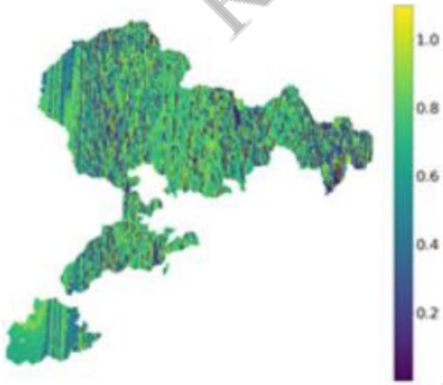
- Learn poacher behavior from data
- Ranger-poacher game to plan patrols
- Deployed in Uganda, China, Malaysia
- Increased detection of poaching
- Available to more than 600 sites worldwide

Data from past patrols
& satellite imagery

Machine Learning Methods

Ensemble Learning, Decision Trees, Neural Networks, Gaussian Process, Markov Random Field, …

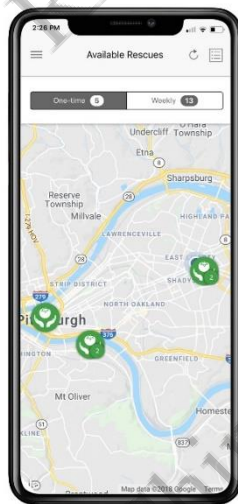
Predicted poaching threat



IJCAI-15, IAAI-16, AAMAS-17, ECML-PKDD 2017, COMPASS 2019, IAAI 2021

In collaboration with Uganda Wildlife Authority, Wildlife Conservation Society, World Wild Fund for Nature, Panthera, Rimba

Improve Efficiency for Food Rescue Platform



Predict whether a rescue will be claimed by a specific volunteer

Rescue i

Volunteer j

Feature

Neural network with 4 hidden layers

Probability that volunteer j claims rescue i

Deployed by 412 Food Rescue

Outline

- Opportunities and Challenges in Applying Multi-Agent Reinforcement Learning
  - MARL for Security and Sustainability
  - Interpretable MARL

- Discussion and Summary

Basic Security Game Model

- N targets
- r (<N) defender resources, each can cover one target
- Attacker choose one target to attack
- Randomized defender strategy
- Strong Stackelberg Equilibrium
  - Coincide with Nash Equilibrium when zero-sum
- Used for security and sustainability problems
- Solved through mathematical programming

Adversary

Defender

		Target #1	Target #2
55.6%	Target #1	5, -3	-1, 1
44.4%	Target #2	-5, 4	2, -1

MARL for Security and Sustainability

- MARL can help tackle more complex scenarios in security and sustainability
  - Patrol with real-time information
  - Robust sequential patrol planning
  - Repeated interaction with unknown attacker
  - Patrol in continuous area

Patrol with Real-Time Information

- Rangers and poachers react to real-time information
- Model the sequential interaction as a Markov game

Footprints

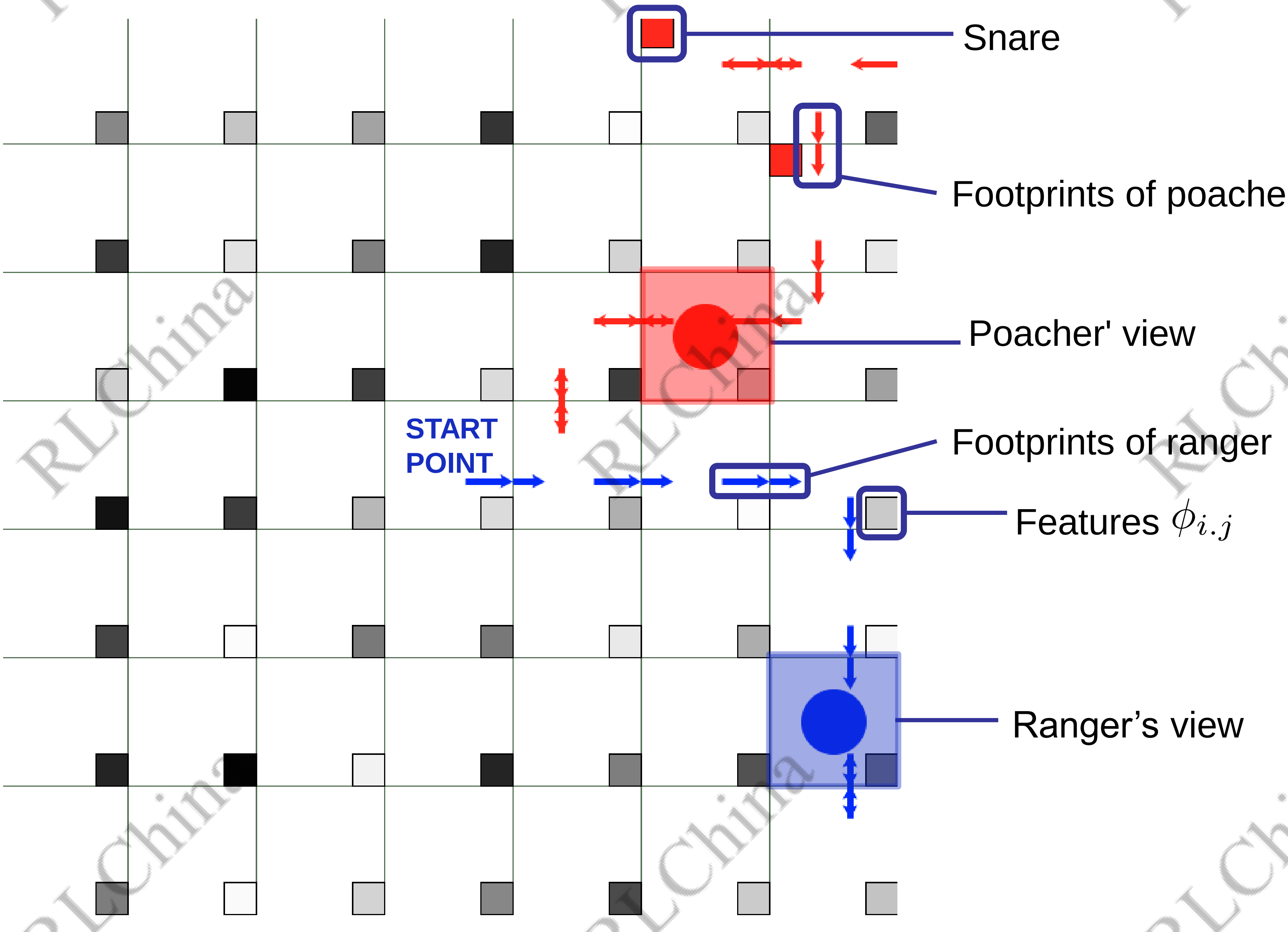
Lighters

Poacher camp

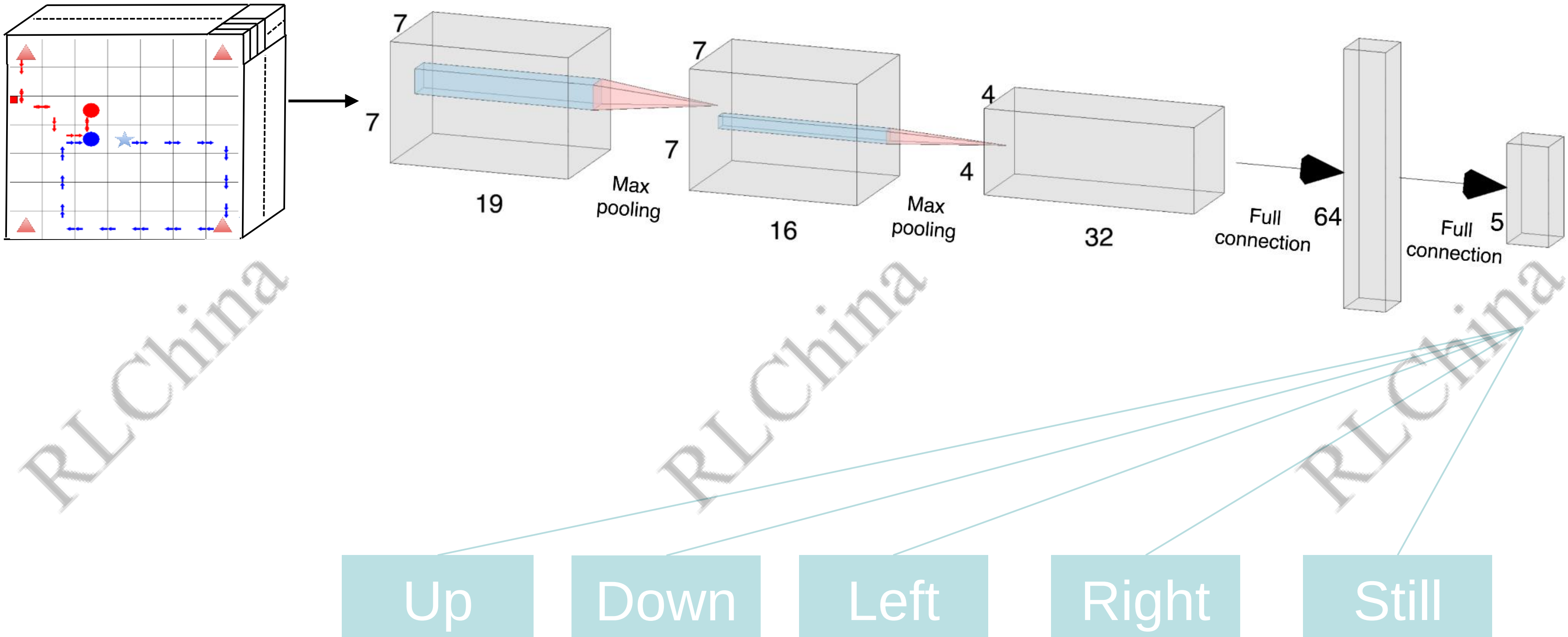
Tree marking

Deep Reinforcement Learning for Green Security Games with Real-Time Information Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, Fei Fang In AAAI-19

Markov Game Formulation



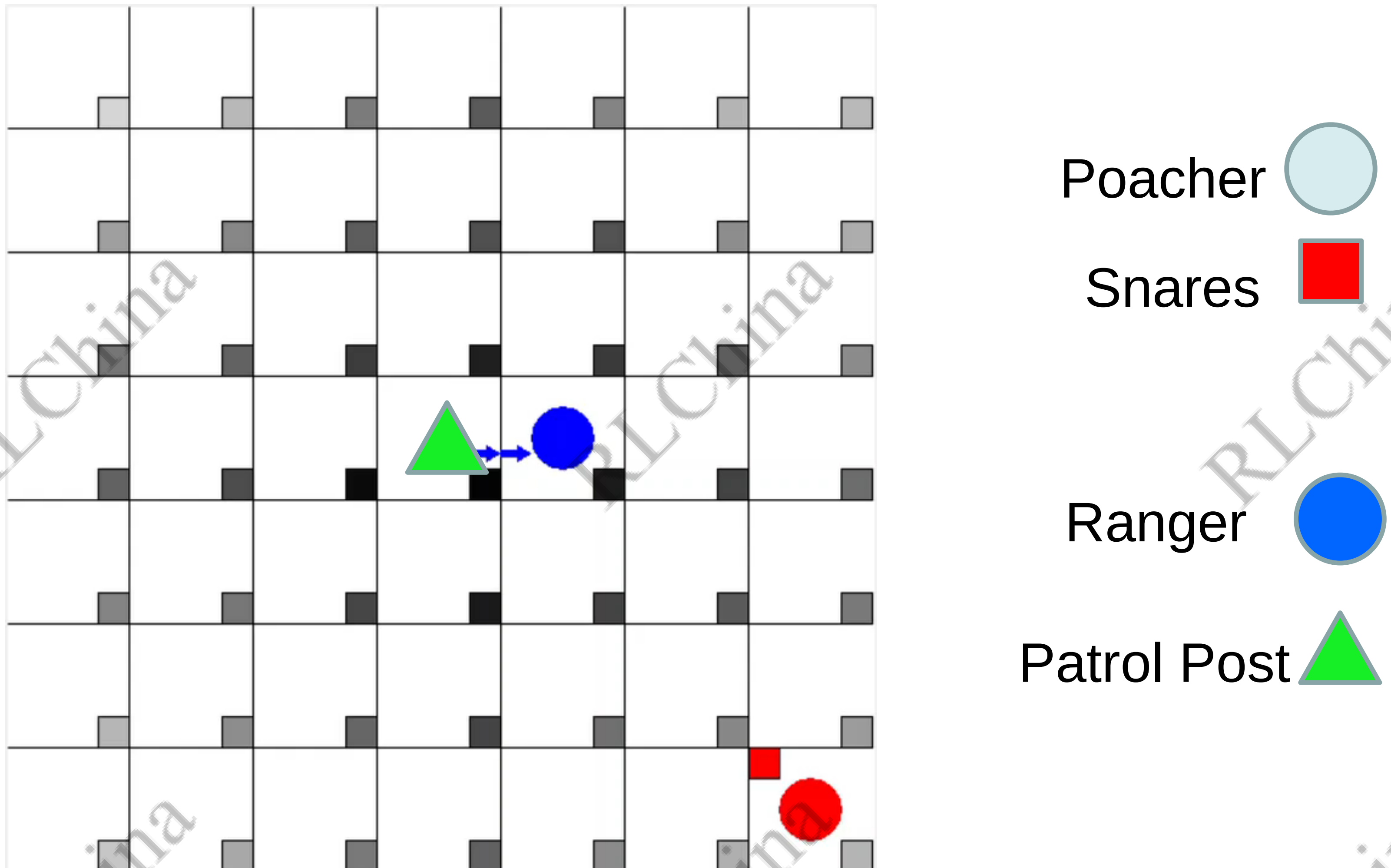
Deep Q Network Trained Against Heuristic Poacher



- Deep Q Network (DQN): Game state → Q-value

Deep Reinforcement Learning for Green Security Games with Real-Time Information Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, Fei Fang In AAAI-19

Deep Q Network Trained Against Heuristic Poacher



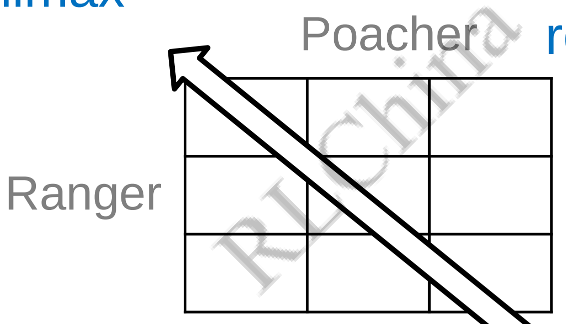
Approximate Equilibrium: DQN + Double Oracle

Compute $\sigma^d, \sigma^a = Nash(G^d, G^a)$

Compute Nash/Minimax

Train $f^d = DQN(\sigma^a)$

Train a new DQN that best responds to poacher's strategy



Train $f^a = DQN(\sigma^a)$

Train a new DQN that best responds to ranger's strategy

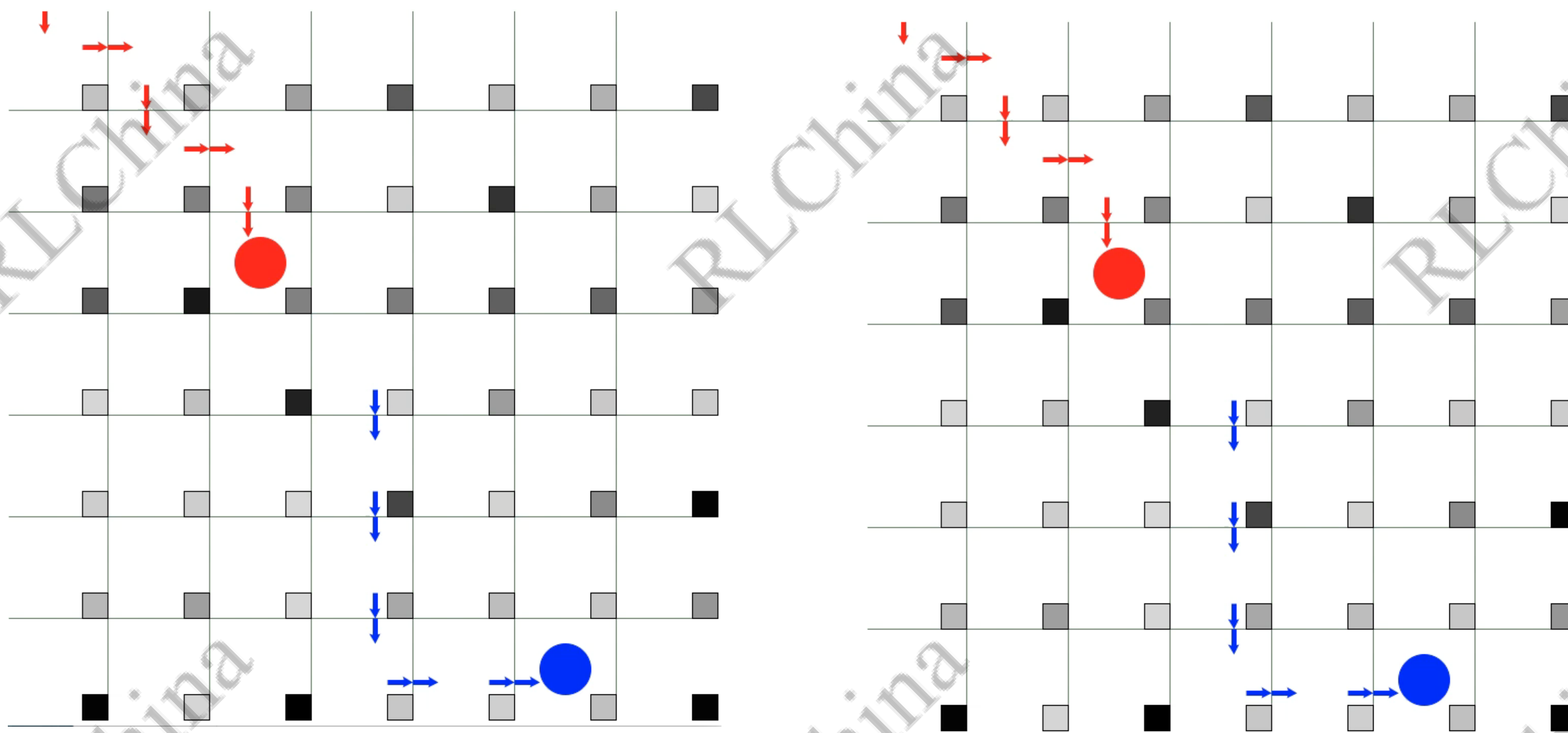
Add f^d, f^a to G^d, G^a

Update sets of DQNs

Deep Reinforcement Learning for Green Security Games with Real-Time Information Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, Fei Fang In AAAI-19

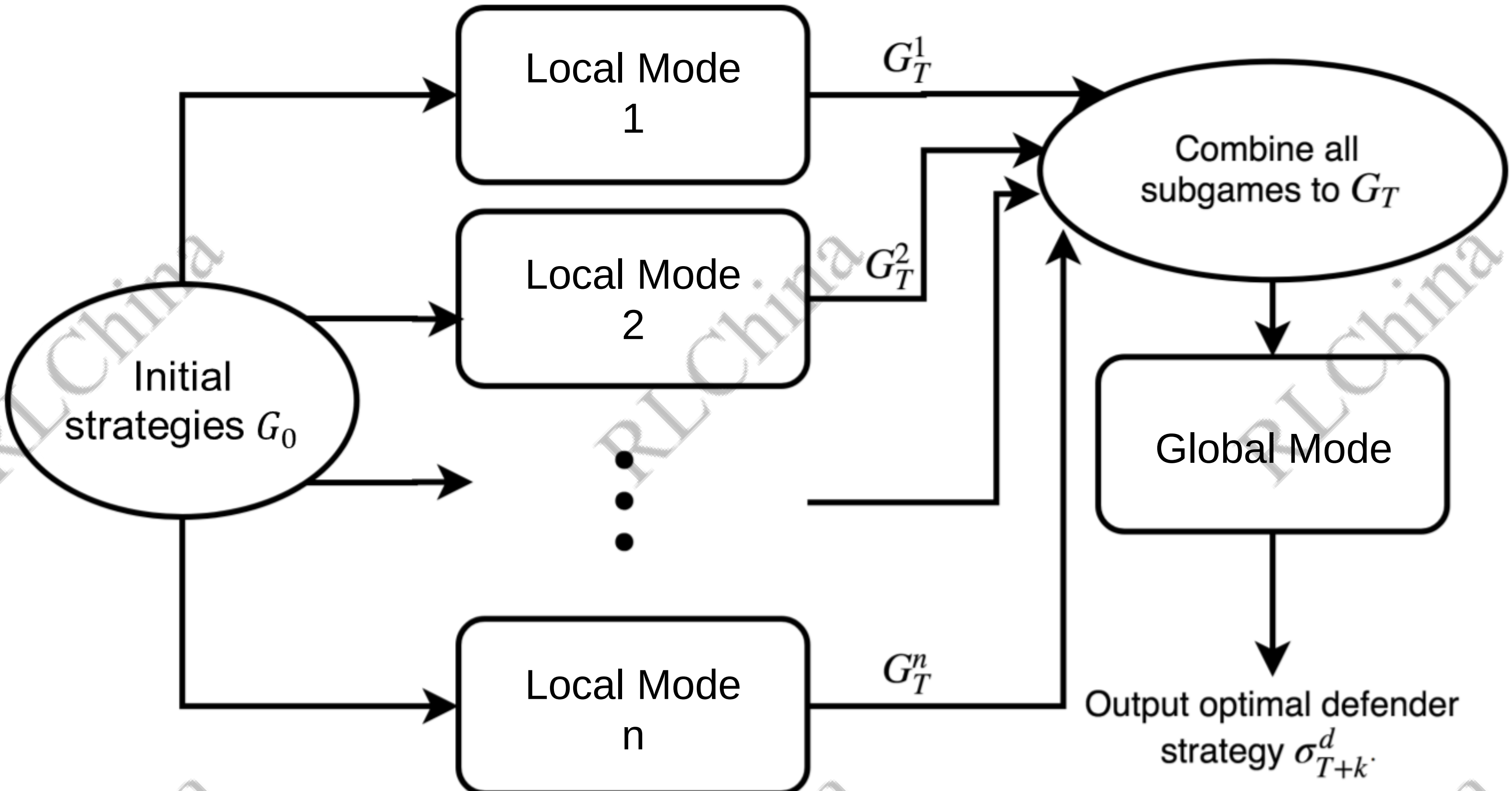
Enhancements

- Use local modes for efficient training
- Start with domain-specific heuristic strategies



Deep Reinforcement Learning for Green Security Games with Real-Time Information Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, Fei Fang In AAAI-19

DeDOL Framework



Deep Reinforcement Learning for Green Security Games with Real-Time Information Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, Fei Fang In AAAI-19

Experiments

- More scalable than counterfactual regret minimization (CFR)
- Better solution quality than vanilla PSRO

	Random Sweeping	Vanilla PSRO	DeDOL Pure Global Mode	DeDOL Local + Global Mode	DeDOL Pure Local Mode	CFR
3×3 Random	-0.04	0.65 (16)	0.73 (16)	**0.85** (10 + 2)	0.71 (20)	**1.01** (3500)
3×3 Gaussian	-0.09	0.52 (16)	0.75 (16)	**0.86** (10 + 2)	0.75(20)	**1.05** (3500)
5×5 Random	-1.91	-8.98 (4)	-1.63 (4)	-0.42 (4 + 1)	**-0.25** (5)	-
5×5 Gaussian	-1.16	-9.09 (4)	-0.43 (4)	**0.60** (4 + 1)	-2.41 (5)	-
7×7 Random	-4.06	-10.65 (4)	-2.00 (4)	**-0.54** (3 + 1)	-1.72(5)	-
7×7 Gaussian	-4.25	-10.08 (4)	-4.15 (4)	**-2.35** (3 + 1)	-2.62(5)	-

Deep Reinforcement Learning for Green Security Games with Real-Time Information Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, Fei Fang In AAAI-19

MARL for Security and Sustainability

- MARL can help tackle more complex scenarios in security and sustainability
  - Patrol with real-time information
  - Robust sequential patrol planning
  - Repeated interaction with unknown attacker
  - Patrol in continuous area

Robust sequential patrol planning

- Data from Queen Elizabeth National Park in Uganda:
  - Poachers are not perfectly rational!
  - Patrol strategies used now will have impact on poachers' behavior in the future!
- Furthermore, wildlife distribution might change due to past poaching activities and impact future poaching

- Learning Poacher Behavior Model + Sequential patrol planning

Robust Reinforcement Learning Under Minimax Regret for Green Security. Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, Milind Tambe In UAI-21

Robust sequential patrol planning

- Uncertainty in Poacher's Behavior Model
  - Poacher's behavior model learned from data is imperfect
  - Poacher's behavior model might change
- Sequential planning → Robust sequential planning

Robust Reinforcement Learning Under Minimax Regret for Green Security. Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, Milind Tambe In UAI-21

Model

- N targets
- Patrol policy $\pi: S \to A$
  - State s_t = (previous actions & wildlife density, current time)
  - Action $a_t \in [0,1]^N$ describes how much patrol effort will be spent on each target
- Poacher behavior model known, with uncertain parameters $\boldsymbol{z} \in Z$ where Z is the uncertainty region
- Goal: learn a minimax regret policy π

$$\min_{\pi} \max_{\mathbf{z}} \left(r(\pi^\star(\mathbf{z}), \mathbf{z}) - r(\pi, \mathbf{z}) \right)$$

Robust Reinforcement Learning Under Minimax Regret for Green Security. Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, Milind Tambe In UAI-21

Model

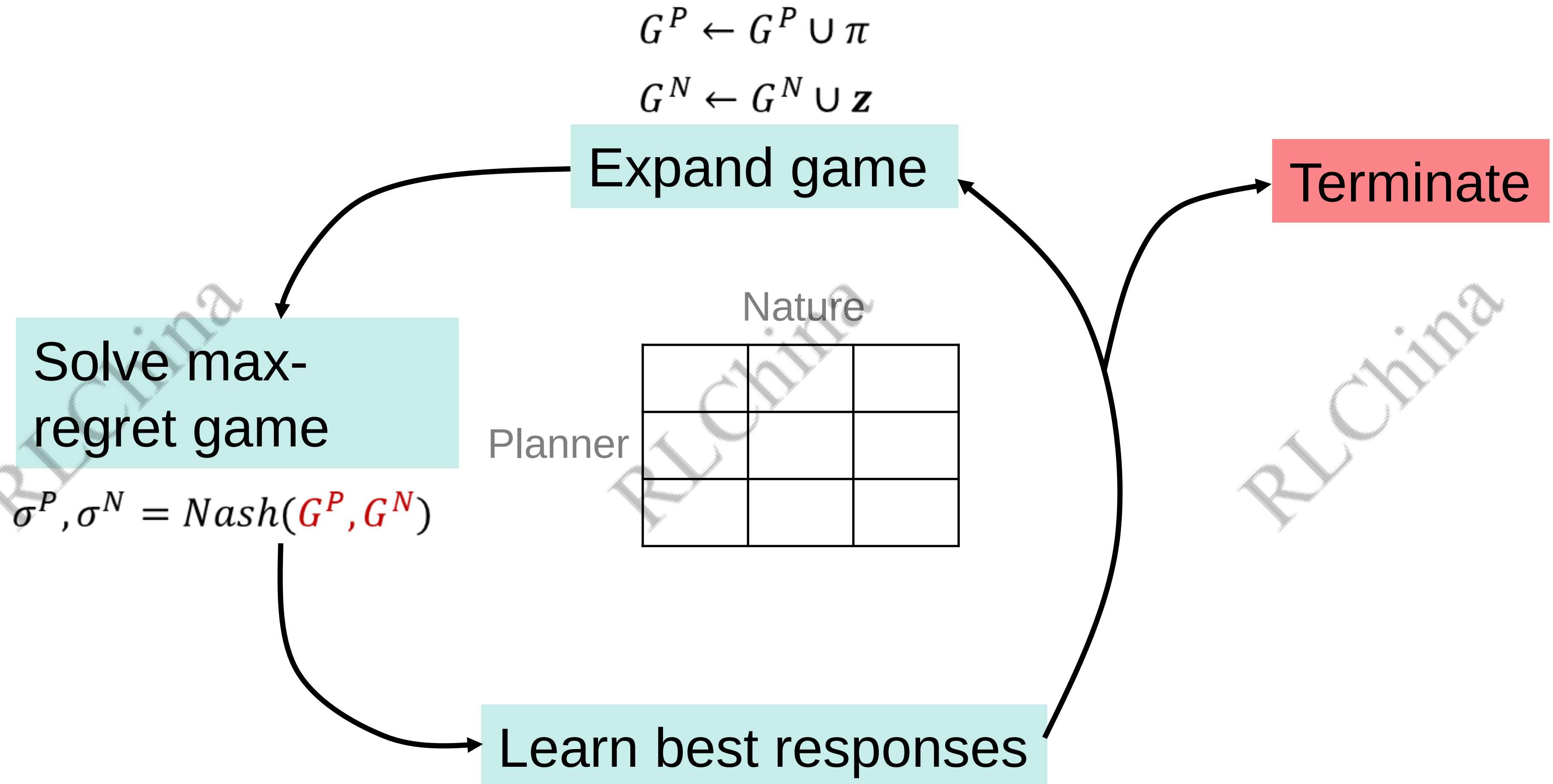
$$\min_{\pi} \max_{\mathbf{z}} \left(r(\pi^\star(\mathbf{z}), \mathbf{z}) - r(\pi, \mathbf{z}) \right)$$

- Alternative View: Zero-sum game between planner and Nature
  - Planner: choose π
  - Nature: choose $\boldsymbol{z}$ & alternative policy $\hat{\pi}$

- Any deterministic choice can be easily exploited
- Need randomized policy

Robust Reinforcement Learning Under Minimax Regret for Green Security. Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, Milind Tambe In UAI-21

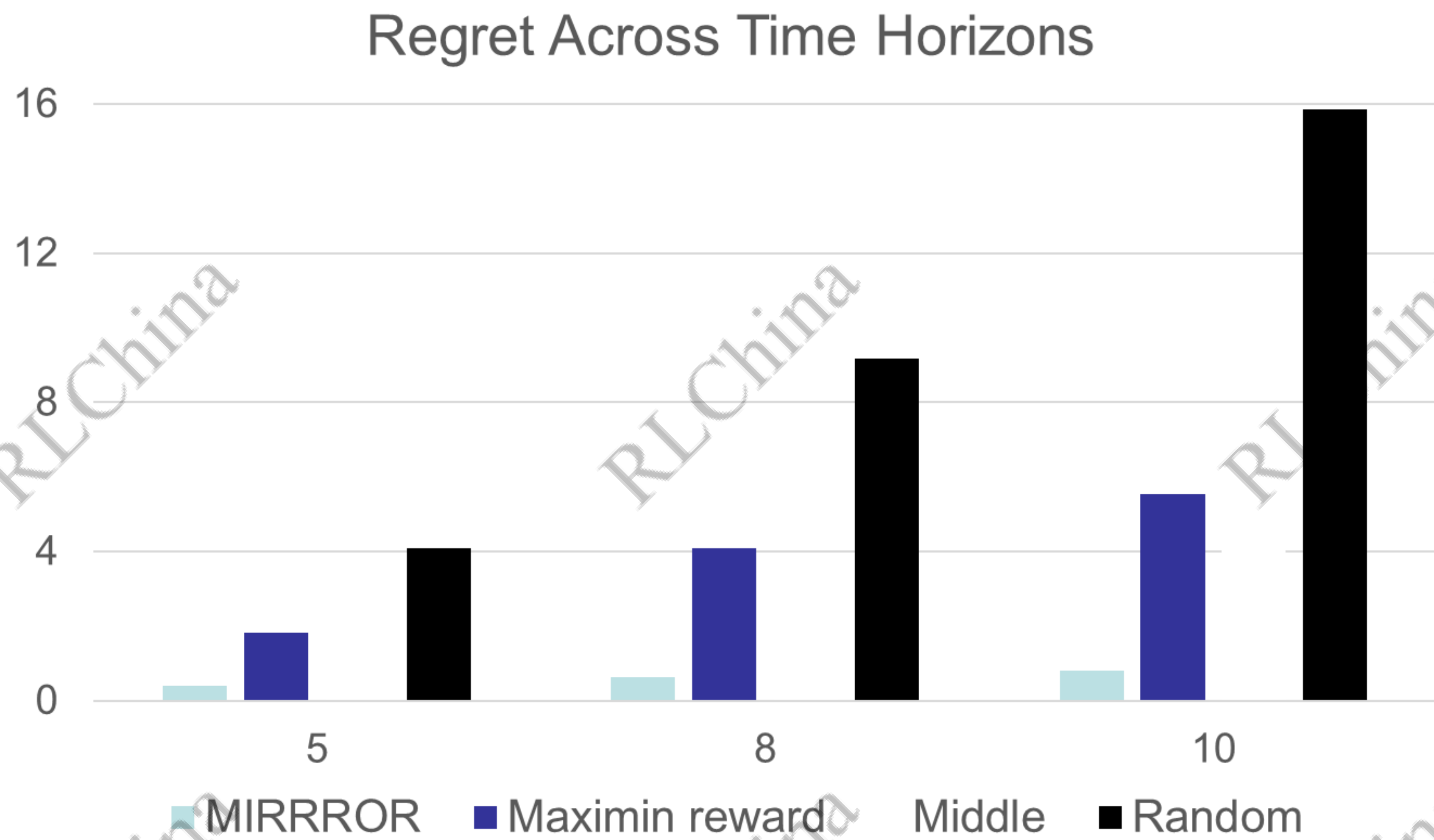
MIRROR – A Double Oracle Approach



Train $\pi = DDPG(\sigma^N)$ Train $\boldsymbol{z}, \hat{\pi} = DDPG(\sigma^P)$

Robust Reinforcement Learning Under Minimax Regret for Green Security. Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, Milind Tambe In UAI-21

Experiments



Robust Reinforcement Learning Under Minimax Regret for Green Security. Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, Milind Tambe In UAI-21

MARL for Security and Sustainability

- MARL can help tackle more complex scenarios in security and sustainability
  - Patrol with real-time information
  - Robust sequential patrol planning
  - Repeated interaction with unknown attacker
  - Patrol in continuous area

Repeated Interaction with Unknown Attacker

- Attackers in cyber security: diverse, with varying sophistication and intent unknown to the defender
- In repeated interaction, defender can infer from attacker's previous actions and plan defense
- Attacker plans actions carefully to maintain informational advantage
- Sequential decision making on both sides!

Temporal Induced Self-Play for Stochastic Bayesian Games. Weizhe Chen, Zihan Zhou, Yi Wu, and Fei Fang. In IJCAI-21

Finitely Repeated Bayesian Security Game Model

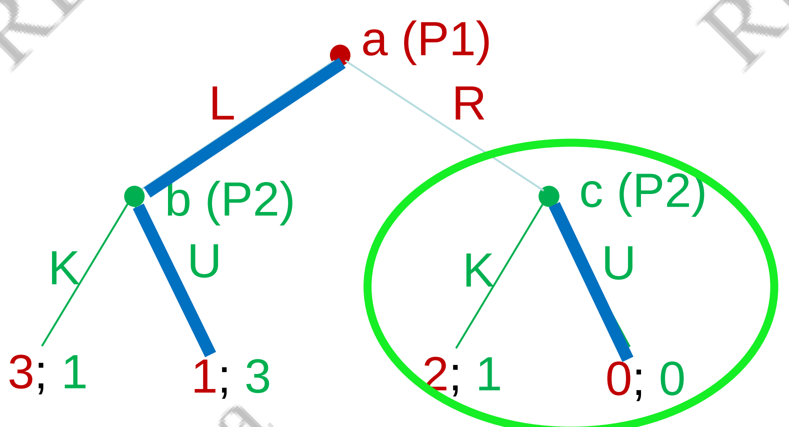
- T round game
- Defender: allocate K resources to N targets in each round
- Attacker: attack a target

Bayesian
  - Reward for successfully attacking a target is determined by type λ
  - Exact attacker type unknown to defender
  - Type distribution is public information

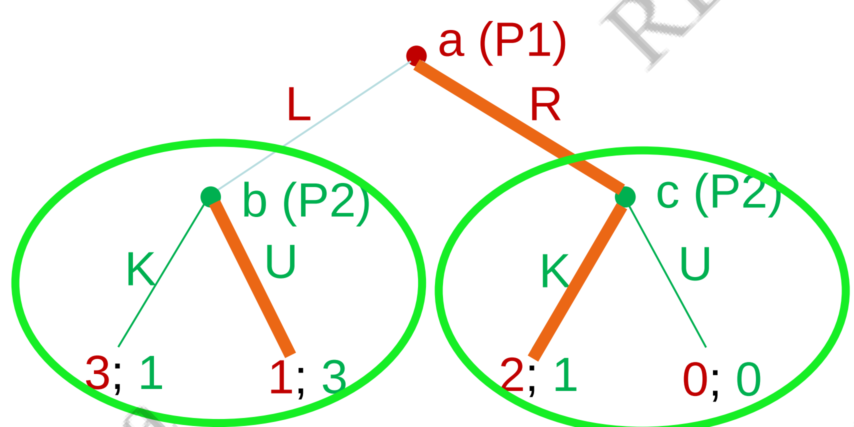
- What strategy should defender use?

Equilibrium Refinement

- There might exist many Nash Equilibria in a game
- Equilibrium Refinement: Get a “desirable” subset of NEs
- Example: Subgame perfect equilibrium
  - Ensure optimality from any point onward



NE



NE and Perfect NE

Bayesian Equilibrium

- Solution Concept for Bayesian games
- Ensure
  - Rationality
  - Belief Consistency: the belief is updated followed the Bayes' rule

Perfect Bayesian Equilibrium

- Perfect Bayesian Equilibrium
  - Equilibrium refinement for Bayesian Equilibrium
  - Extends Perfect NE to Bayesian games
  - Sequential rationality starting from **any** information set

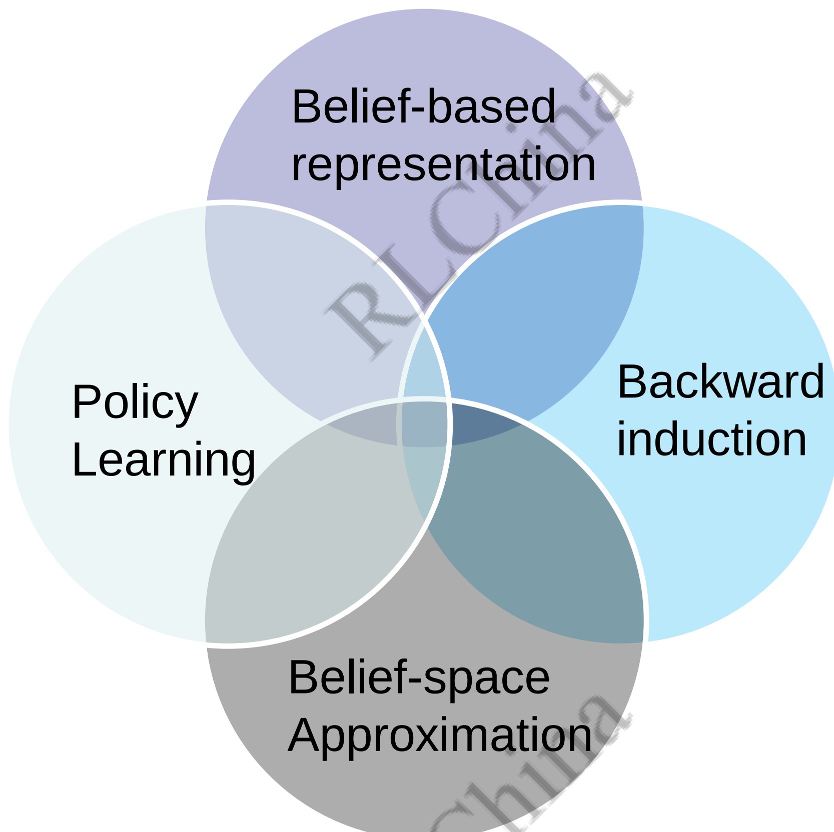
- Our goal: Find PBE for Finitely Repeated Bayesian Security Game

- Most existing work solve using Mathematical Programming-based method (Nguyen et al. 2019[1]; Guo et al. 2017[2])
  - Very precise
  - Lacks scalability: long time and large memory to solve

[1] Nguyen et al. Deception in finitely repeated security games. In *AAAI-19*
[2] Guo et al. Comparing Strategic Secrecy and Stackelberg Commitment in Security Games. In IJCAI-17

An RL Approach: Temporal Induced Self-Play

- Temporal Induced Self-Play (TISP)
  - A framework that can be combined with different learning algorithms



Belief-based representation

- Use belief instead of history: $\pi(s, b)$ instead of $\pi(h)$
  - $\pi(\text{attack Target 1 in } (l-1) \text{ round, 2 in } (l-2) \text{ round,} ..)$ is now $\pi(\text{0.2 prob. of being attacker type a})$

- Helps in the case with long history

Backward Induction

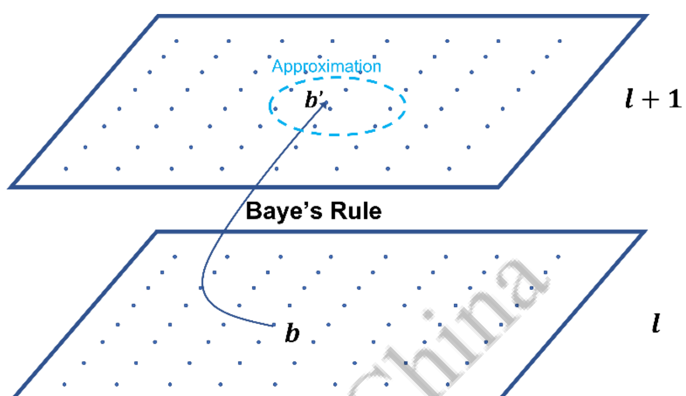
- Reverse the training process
  - From round $L-1$ to round $L-2$, to …, to round 0
  - Use trained value network V and policy network π in round $l+1$ when training round l
    - Do not sample the whole trajectory from round 0 to round $L-1$, but one step trajectory from round l to round $l+1$.
    - Using a special reset function to help

Belief Space Approximation

- Training: Sample K belief vectors, and train the strategies specifically conditioning on the belief and round,
- Query:

$$\pi(a|b,s) = \frac{\sum_{k=1}^{K} \pi_{\theta_k}(a|s;b_k)\, w(b,b_k)}{\sum_{k=1}^{K} w(b,b_k)}$$

where $w(b,b') = 1/\max\{\epsilon, \|b-b'\|^2\}$



Policy Learning

- Policy gradient:
  - Update rule changed:

$$\nabla_\theta V^\lambda(\pi, b, s) = \sum_{a \in A} \nabla_\theta \left(\pi_\theta (a \mid b, s) Q^\lambda(\pi, b, s, a)\right)$$

$$= E[\, Q^\lambda(\pi, b, s, a) \nabla_\theta \, ln\pi_\theta(a \mid b, s) + \boxed{\gamma \nabla_\theta b' \nabla_{b'} V^\lambda(\pi, b', s')}]$$

- Regret matching:

$$\pi^{t+1}(a|s, b) = \frac{\left(R^{t+1}(s,b,a)\right)^+}{\sum_{a'} (R^{t+1}(s,b,a'))^+}$$

where

$$R^{t+1}(s, b, a) = \sum_{\tau=1}^{t} Q^\tau(\pi^\tau, s, b, a) - V_\phi^\tau(\pi^\tau, s, b)$$

Temporal Induced Self-play Training

Algorithm 1 Temporal-Induced Self-Play

---

for $l = L-1, \ldots, 0$ **do**
    for $k = 1, 2, \ldots K$ **do** $\triangleright$ run in parallel
        for $t = 1, \ldots, T$ **do**
            Initialize replay buffer $D = \{\}$ and π^0
            for $j = 1, \ldots,$ batch size **do** $\triangleright$ parallel
                $s \leftarrow sub\_reset(l, b_k)$;
                $a \leftarrow \pi^{t-1}_{\theta_{l,k}}(s; b_k)$;
                get next state s' and utility u from env;
                $D \leftarrow D + (s, a, s', u)$;
            Update $V^t_{\phi_{l,k}}$ and $\pi^t_{\theta_{l,k}}$ using D;
        $V_{\phi_{l,k}} \leftarrow V^n_{\phi_{l,k}}$, $\pi_{\theta_{l,k}} \leftarrow \pi^n_{\theta_{l,k}}$
return $\{\pi_{\theta_{l,k}}, V_{\phi_{l,k}}\}_{0 \le l < L, 1 \le k \le K}$;

Test-time Policy Transformation

Algorithm 2 Compute Test-Time Strategy

1: **function** GETSTRATEGY($h^l, \pi_{\theta_1}, \ldots, \pi_{\theta_L}$)
2: $b^0 \leftarrow p^0$
3: **for** $j \leftarrow 0, \ldots, l-1$ **do**
4: update b^{j+1} using b^j, s^j, a^j and π_{θ_j} with

$$b_\lambda^{l+1} = \frac{\pi_{1,l}\left(a_1^l | s^l, b_\lambda^l, \lambda\right) b_\lambda^l}{\sum_{\lambda' \in \Lambda} \pi_{1,l}\left(a_1^l | s^l, b_{\lambda'}^l, \lambda'\right) b_{\lambda'}^l}$$

5: **return** $\pi(a|b^l, s^l)$ with

$$\pi(a|b,s) = \frac{\sum_{k=1}^K \pi_{\theta_k}(a|s; b_k) w(b, b_k)}{\sum_{k=1}^K w(b, b_k)}$$

Experiment: Grid-World Games

- Can solve larger repeated security games

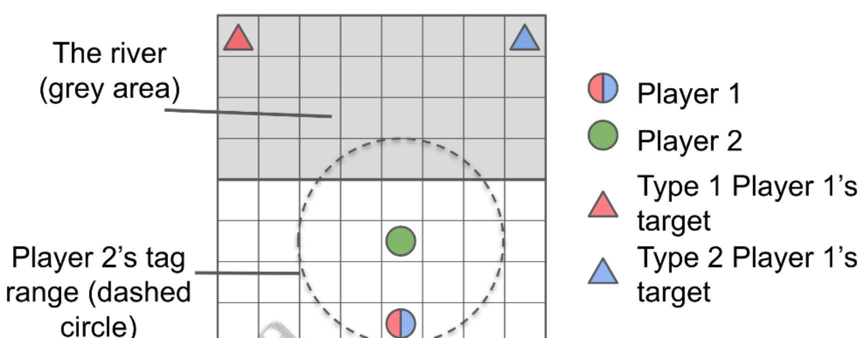
L	2	4	6	8	10
MP	$\approx 10^{-8}$	$\approx 10^{-6}$	$\approx 10^{-5}$	N/A	N/A
TISP-PG	0.053	0.112	0.211	0.329	0.473
TISP-CFR	0.008	0.065	0.190	0.331	0.499

(a) $|A| = 2$

L	2	4	6	8	10
MP	$\approx 10^{-6}$	$\approx 10^{-6}$	$\approx 10^{-3}$	N/A	N/A
TISP-PG	0.120	0.232	0.408	0.599	0.842
TISP-CFR	0.002	0.049	0.285	0.525	0.847

(b) $|A| = 5$

- Can generalize to grid-world games
  - Much higher quality than other learning-based method



	TISP-PG	RNN	BPG
P2 reward	-1.90	-1.67	-0.98
P1 reward (ally)	-2.55	-2.87	-3.26
P1 reward (enemy)	-2.41	-2.71	-9.29

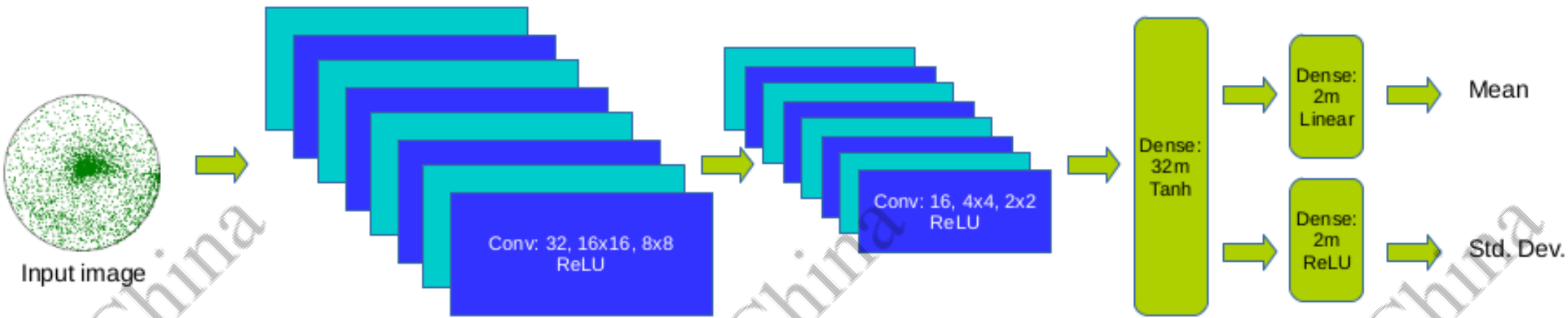
Agents' reward when P2 (approx.) best responding to P1's trained policy. Higher P1 reward, lower P2' reward is better

MARL for Security and Sustainability

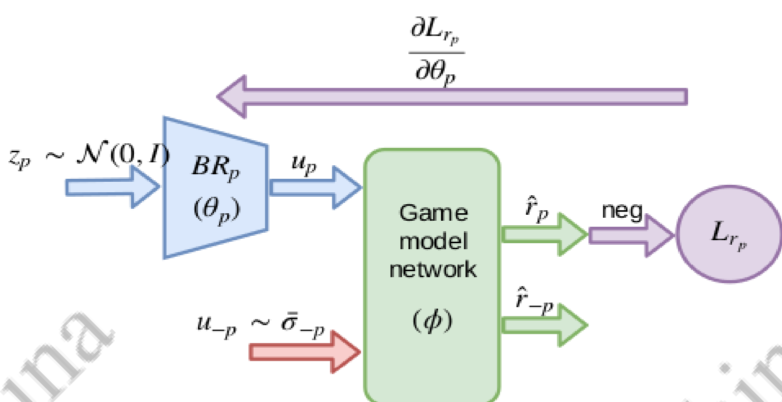
- MARL can help tackle more complex scenarios in security and sustainability
  - Patrol with real-time information
  - Robust sequential patrol planning
  - Repeated interaction with unknown attacker
  - Patrol in continuous area

Patrol in Continuous Area: Combat Illegal Logging

OptGradFP: CNN + Fictitious Play



DeepFP: Generative network + Fictitious Play



Policy Learning for Continuous Space Security Games using Neural Networks. Nitin Kamra, Umang Gupta, Fei Fang, Yan Liu, Milind Tambe. In AAAI-18

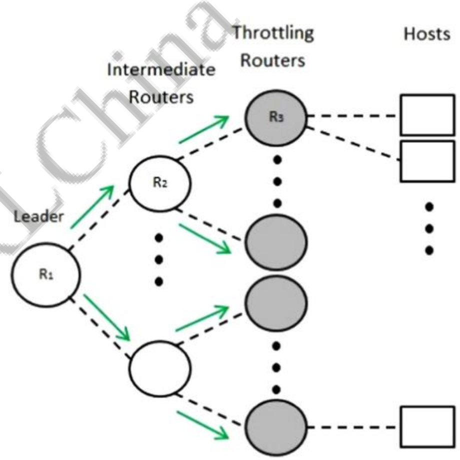
DeepFP for Finding Nash Equilibrium in Continuous Action Spaces. Nitin Kamra, Umang Gupta, Kai Wang, Fei Fang, Yan Liu, Milind Tambe. In GameSec-19

Outline

- Opportunities and Challenges in Applying Multi-Agent Reinforcement Learning
    - MARL for Security and Sustainability
    - Interpretable MARL
- Discussion and Summary

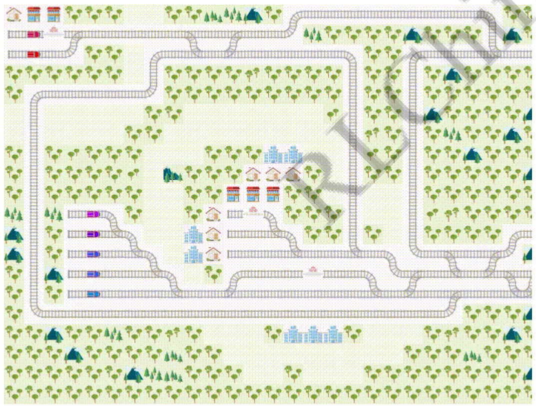
MARL for Real-World Applications

Cyber defense
[Malialis & Kudenko, 2015]

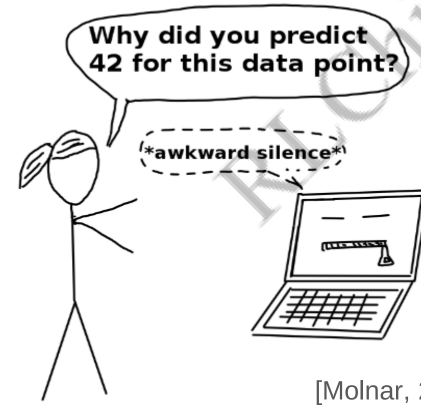


Anti-Poaching
[Wang et al., 2019]

Train scheduling
[Mohanty et al., 2020]



Interpretable RL



[Molnar, 2020]

- Neural networks are difficult to understand
- An increasing interest in interpretable RL!
  - See our recent survey: A Survey of Explainable Reinforcement Learning (https://arxiv.org/abs/2202.08434)

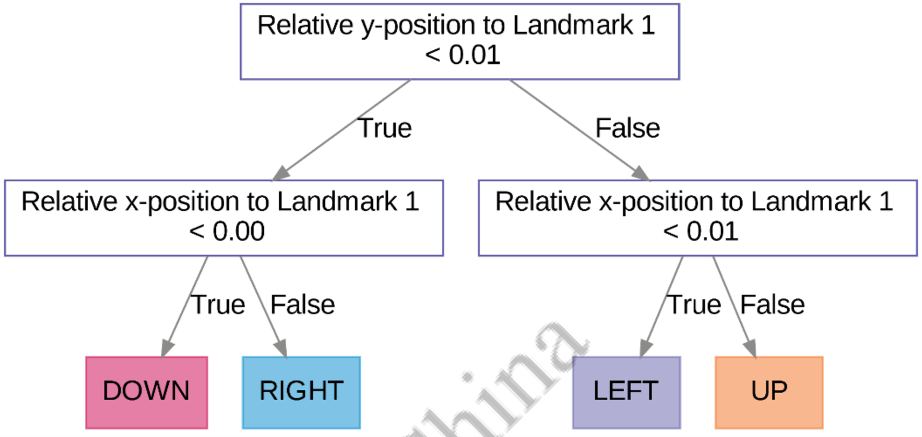
- Need to understand, verify, and predict the behavior of machine learning models for real-world problems

- One interpretable model: decision trees

Decision Trees for Interpretable RL

- Interpretable model family used in reinforcement learning [Lipton, 2018]
    - Simulatable
    - Decomposable
    - Algorithmically transparent

- Used to represent policies [Pyeatt, 2003]



Example decision-tree policy

Interpretable MARL

- Interpretable MARL is underexplored!
- Some work generates explanations from noninterpretable policies
  - Use attention to select and focus on critical factors [Iqbal&Sha, 2019, Li et al. 2019, Motokawa& Sugawara 2021]
  - Generates explanations as verbal explanations with predefined rules or Shapley values [Wang et al. 2020, Heuillet et al., 2022]
- Some approximates noninterpretable MARL policies to interpretable ones
  - Construct argument preference graphs given manually-provided arguments [Kazhdan et al. 2020]

Interpretable MARL

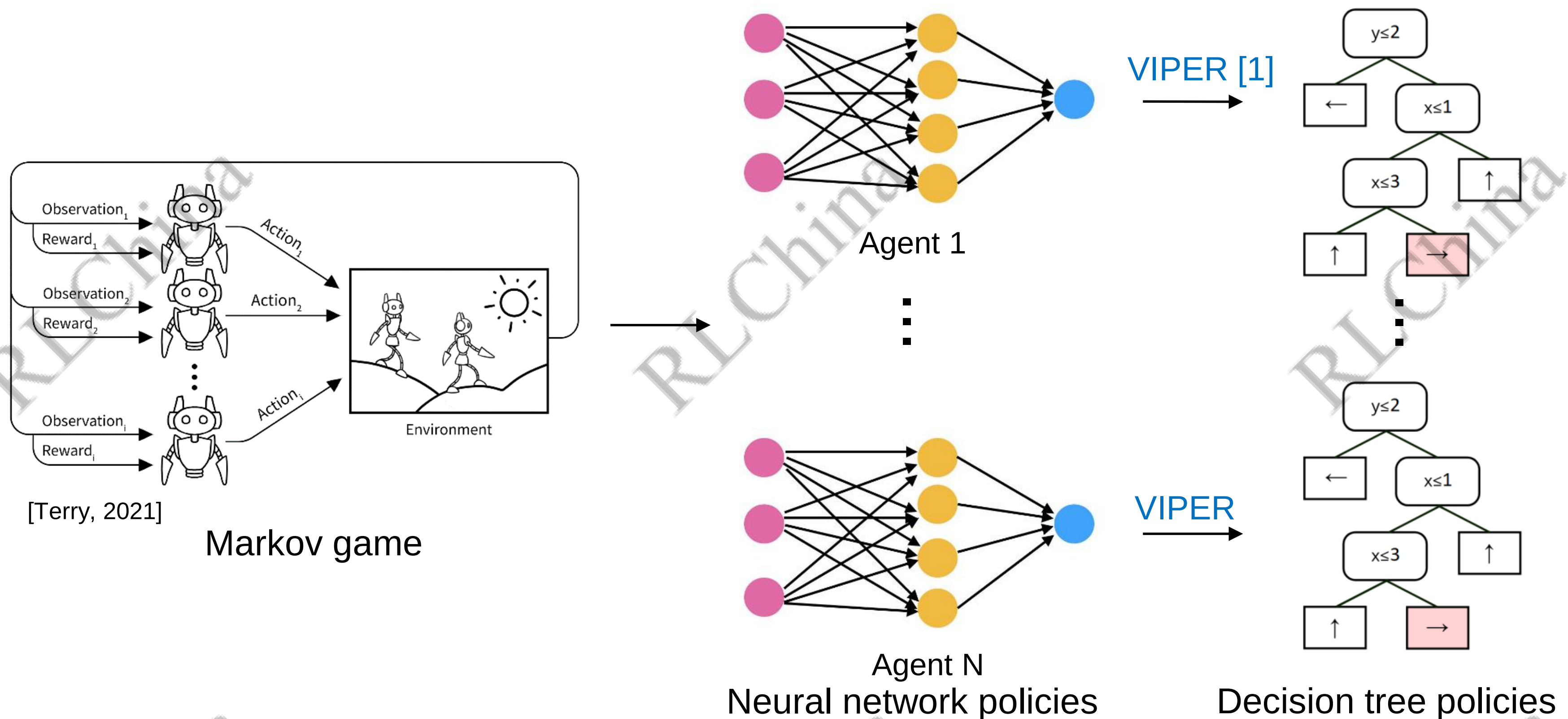
We contribute…

- **IVIPER** and **MAVIPER**, two portable algorithms for learning interpretable decision tree policies in the multi-agent setting
- An analysis of these algorithms on three different environments

MAVIPER: Learning Decision Tree Policies for Interpretable Multi-Agent Reinforcement Learning
Stephanie Milani*, Zhicheng Zhang*, Nicholay Topin, Zheyuan Ryan Shi, Charles Kamhoua,
Evangelos E. Papalexakis, Fei Fang ECML-PKDD, 2022

IVIPER

- Independent VIPER (IVIPER) trains policies independently



MAVIPER: Learning Decision Tree Policies for Interpretable Multi-Agent Reinforcement Learning
Stephanie Milani*, Zhicheng Zhang*, Nicholay Topin, Zheyuan Ryan Shi, Charles Kamhoua, Evangelos E. Papalexakis, Fei Fang ECML-PKDD, 2022

[1] Bastani, O., et al.: Verifiable reinforcement learning via policy extraction. In: NeurIPS (2018)

IVIPER

Algorithm 1 IVIPER in Multi-Agent Setting

Input: (X, A, P, R), π^* , $Q^{\pi^*} = (Q_1^{\pi^*}, ..., Q_N^{\pi^*})$, K, M
Output: $\hat{\pi}_1, ..., \hat{\pi}_N$

for i=1 to N **do**
  Initialize dataset $\mathcal{D}_i \leftarrow \emptyset$ and policy $\hat{\pi}_i^0 \leftarrow \pi_i^*$
  for $m = 1$ to M **do**
    Sample K trajectories: $\mathcal{D}_i^m \leftarrow \{(x, \pi_1^*(o_1), ..., \pi_N^*(o_N)) \sim d^{\hat{\pi}_i^{m-1}, \pi_{-i}^*}\}$
    Aggregate dataset $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \mathcal{D}_i^m$
    Resample dataset according to loss:
      $\mathcal{D}_i' \leftarrow \{(x, \overrightarrow{a}) \sim p((x, \overrightarrow{a})) \propto \tilde{l}_i(x)\mathbb{I}[(x, \overrightarrow{a}) \in \mathcal{D}_i]\}$
    Train decision tree $\hat{\pi}_i^m \leftarrow \text{TrainDecisionTree}(\mathcal{D}_i')$
  Get best policy $\hat{\pi}_i \leftarrow \text{BestPolicy}(\hat{\pi}_i^1, ..., \hat{\pi}_i^M, \pi_{-i}^*)$
return Best policies for each agent $\hat{\pi} = (\hat{\pi}_1, ..., \hat{\pi}_N)$

IVIPER

Good news and bad news…

- Independent learning is parallelizable!
- But it ignores coordination…

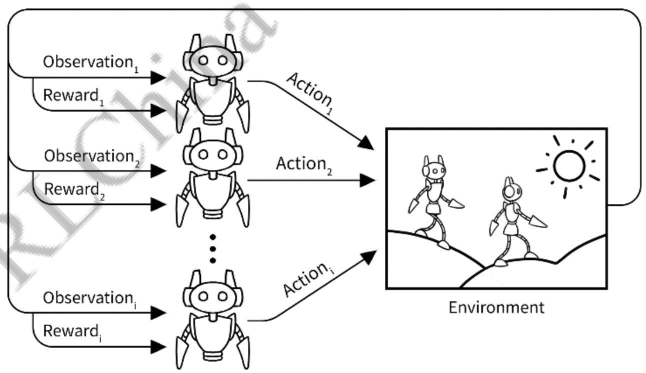
IVIPER

- How to address the issue of coordination in IVIPER?

- Emphasize states where coordination matters
  - A bad move of one agent can impact greatly all agents
  - Jointly learn the decision trees
- Joint accuracy sometimes matters more than individual accuracy
  - When one agent mispredicts, the accurate prediction of other agents is less meaningful
  - Predict whether or not the other agents might mispredict

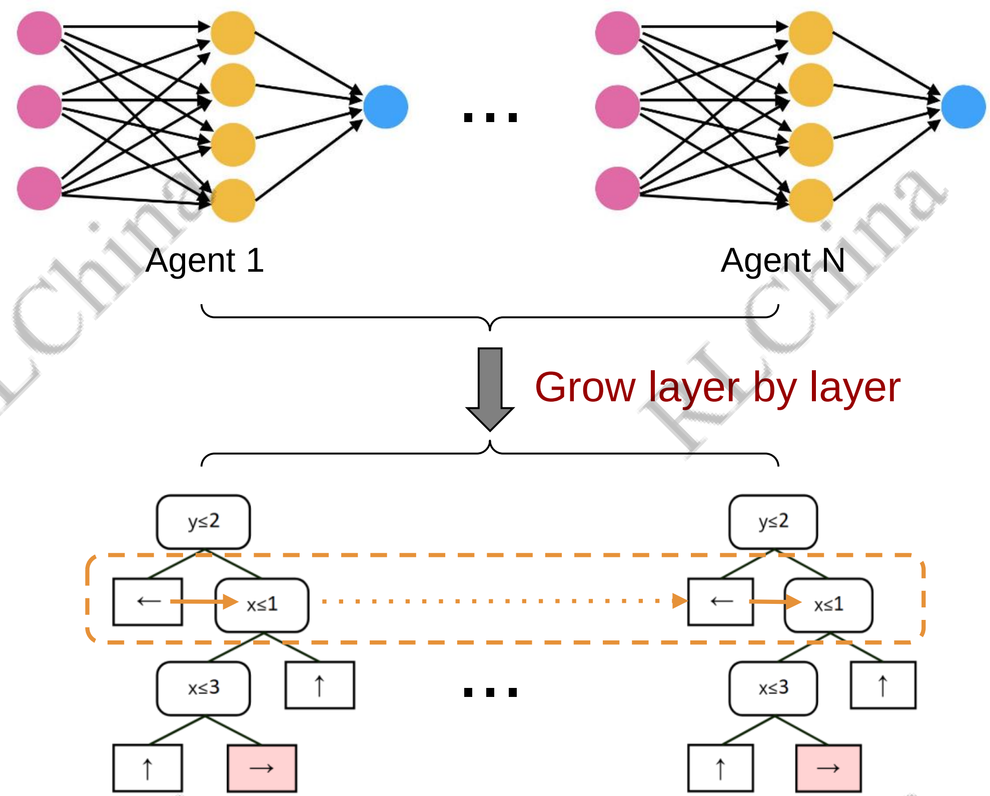
MAVIPER

- Multi-Agent VIPER (MAVIPER) trains policies jointly



[Terry, 2021]

Markov game



Train the decision trees jointly to improve coordination among agents!!

MAVIPER

Algorithm 2 MAVIPER (Joint Training)

Input: $(\mathcal{X}, A, P, R)$, π^*, $Q^{\pi^*} = (Q_1^{\pi^*}, \ldots, Q_N^{\pi^*})$, K, M
Output: $(\hat{\pi}_1, \ldots, \hat{\pi}_N)$

1: Initialize dataset $\mathcal{D} \leftarrow \emptyset$ and policy for each agent $\hat{\pi}_i^0 \leftarrow \pi_i^* \ \forall i \in N$
2: **for** $m = 1$ to M **do**
3: Sample K trajectories: $\mathcal{D}^m \leftarrow \{(x, \pi_1^*(o_1), \ldots, \pi_N^*(o_N)) \sim d^{(\hat{\pi}_1^{m-1}, \ldots, \hat{\pi}_N^{m-1})}\}$
4: Aggregate dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}^m$
5: For each agent i, resample $\mathcal{D}_i$ according to loss:
$\mathcal{D}_i \leftarrow \{(x, \vec{a}) \sim p((x, \vec{a})) \propto \tilde{l}_i(x)\mathbb{I}[(x, \vec{a}) \in \mathcal{D}]\} \forall i \in N$
6: Jointly train DTs: $(\hat{\pi}_1^m, \ldots, \hat{\pi}_N^m) \leftarrow \text{TrainJointTrees}(\mathcal{D}_1, \ldots, \mathcal{D}_N)$
7: **return** Best set of agents $\hat{\pi} = (\hat{\pi}_1, \ldots, \hat{\pi}_N) \in \{(\hat{\pi}_1^1, \ldots, \hat{\pi}_N^1), \ldots, (\hat{\pi}_1^M, \ldots, \hat{\pi}_N^M)\}$

MAVIPER

- MAVIPER incorporates a novel resampling scheme
- Insight: agents should care most about states with a **large gap** between worst-case and expert performance

$$\tilde{l}(x) = \mathbb{E}_{a_{-i}}\left[Q_i^{\pi^*}(x, \pi_i^*(o_i), a_{-i}) - \min_{a_i \in \mathcal{A}_i} Q_i^{\pi^*}(x, a_i, a_{-i})\right]$$

Best case performance of agent i

Worst case performance of agent i

Expectation taken over other the actions of other agents

MAVIPER

Algorithm 2 MAVIPER (Joint Training)

Input: $(\mathcal{X}, A, P, R)$, π^*, $Q^{\pi^*} = (Q_1^{\pi^*}, \ldots, Q_N^{\pi^*})$, K, M
Output: $(\hat{\pi}_1, \ldots, \hat{\pi}_N)$

Initialize dataset $\mathcal{D} \leftarrow \emptyset$ and policy for each agent $\hat{\pi}_i^0 \leftarrow \pi_i^* \ \forall i \in N$
for $m = 1$ to M **do**
    Sample K trajectories: $\mathcal{D}^m \leftarrow \{(x, \pi_1^*(o_1), \ldots, \pi_N^*(o_N)) \sim d^{(\hat{\pi}_1^{m-1}, \ldots, \hat{\pi}_N^{m-1})}\}$
    Aggregate dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}^m$
    For each agent i, resample $\mathcal{D}_i$ according to loss:
    $\mathcal{D}_i \leftarrow \{(x, \vec{a}) \sim p((x, \vec{a})) \propto \tilde{l}_i(x) \mathbb{I}[(x, \vec{a}) \in \mathcal{D}]\} \forall i \in N$
    Jointly train DTs: $(\hat{\pi}_1^m, \ldots, \hat{\pi}_N^m) \leftarrow \mathrm{TrainJointTrees}(\mathcal{D}_1, \ldots, \mathcal{D}_N)$
return Best set of agents $\hat{\pi} = (\hat{\pi}_1, \ldots, \hat{\pi}_N) \in \{(\hat{\pi}_1^1, \ldots, \hat{\pi}_N^1), \ldots, (\hat{\pi}_1^M, \ldots, \hat{\pi}_N^M)\}$

MAVIPER

```
function TrainJointTrees(D_1, ..., D_N)
    Initialize decision trees π̂_1^m, ..., π̂_N^m.
    repeat
        Grow one more level for agent i's tree π̂_i^m ← Build(π̂_1^m, ..., π̂_N^m, D_i)
        Move to the next agent: i ← (i + 1)%N
    until all trees have grown to the maximum depth allowed
    return decision trees π̂_1^m, ..., π̂_N^m
```

function $\textsc{TrainJointTrees}(\mathcal{D}_1, \ldots, \mathcal{D}_N)$

Initialize decision trees $\hat{\pi}_1^m, \ldots, \hat{\pi}_N^m$.

repeat

Grow one more level for agent i's tree $\hat{\pi}_i^m \leftarrow \boxed{\text{Build}(\hat{\pi}_1^m, \ldots, \hat{\pi}_N^m, \mathcal{D}_i)}$

Move to the next agent: $i \leftarrow (i+1)\%N$

until all trees have grown to the maximum depth allowed

return decision trees $\hat{\pi}_1^m, \ldots, \hat{\pi}_N^m$

MAVIPER

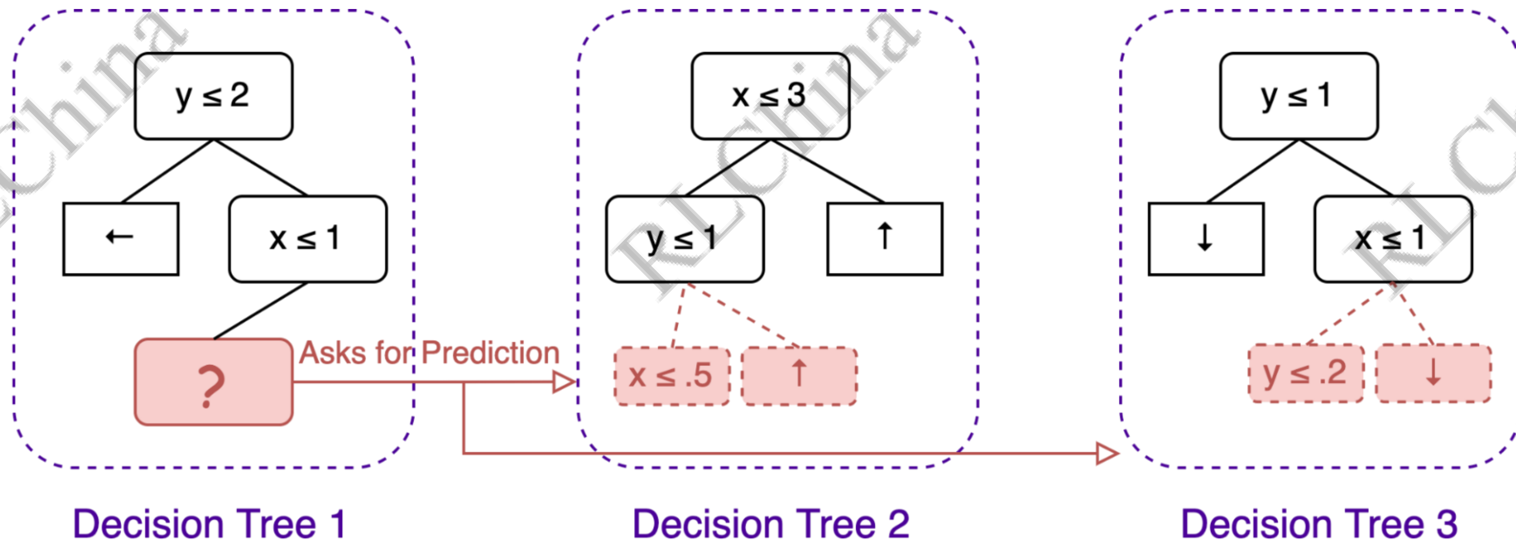
- MAVIPER filters out training data that may not yield much benefit

- Insight: the correct prediction of one agent alone may not yield much benefit if the other agents are incorrect

- For each data point:
  - What action would each decision-tree agent take? Does it align with the expert?
  - Do most agents correctly identify the action to take? If not, remove the data from the dataset.

MAVIPER

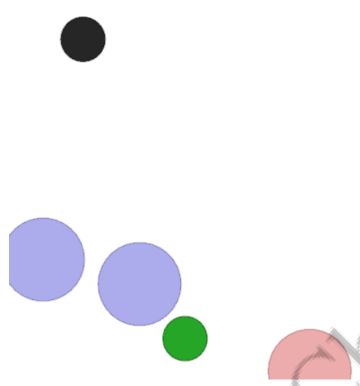
- MAVIPER incorporates a prediction module for joint training



Experiments

- Example evaluation environment: Physical Deception

- **Defenders** spread out to protect targets
- **Attacker** wants to reach green target (but does not know which of two targets is correct)



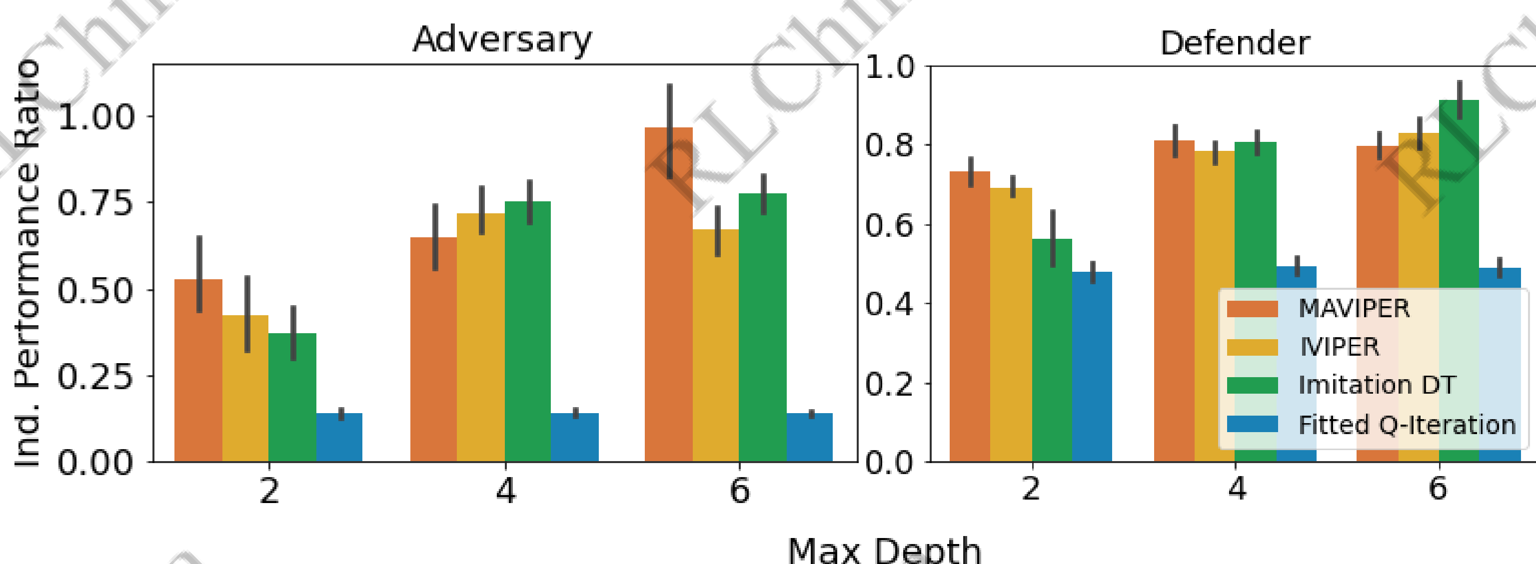
Experiments

- We want high-performing interpretable policies

$$\text{Performance Ratio} = \frac{\text{Performance of the agent when it uses a DT policy}}{\text{Performance of the agent when it uses a NN policy}}$$

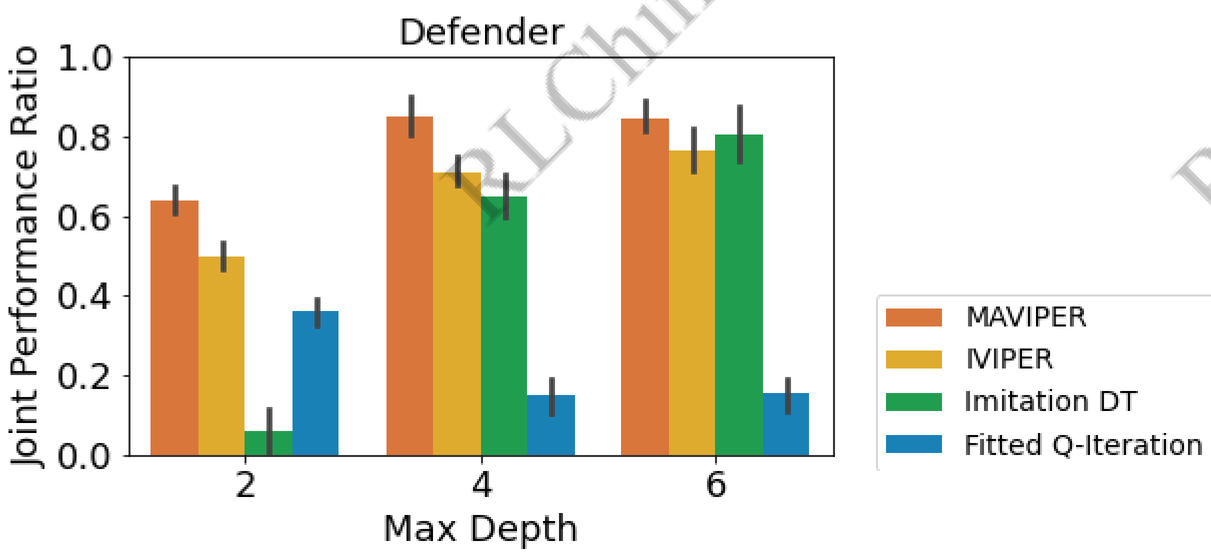
Experiments

- How well do MAVIPER and IVIPER perform when measuring the performance of individual decision-tree agents?



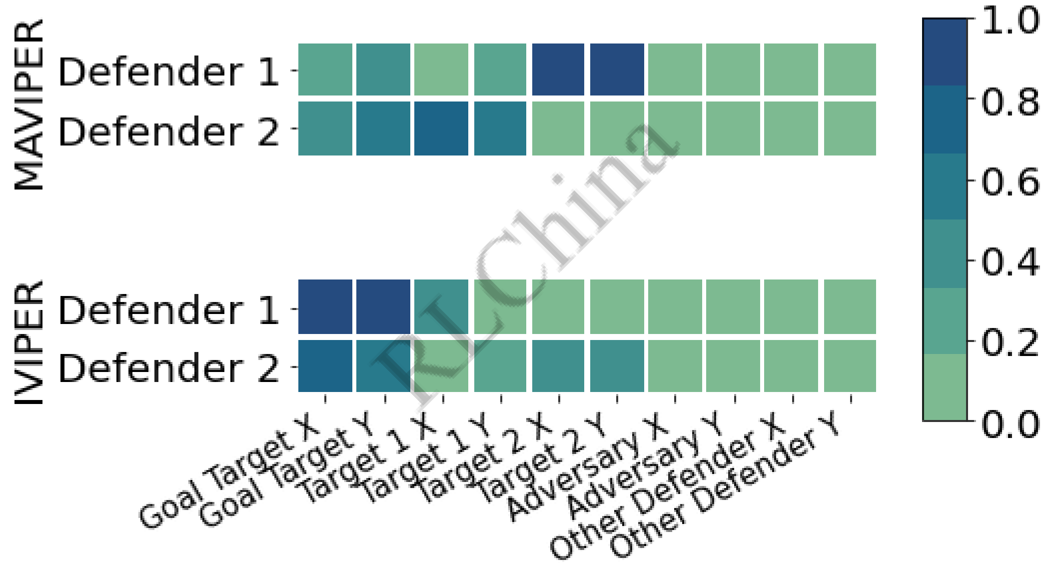
Experiments

- How well do MAVIPER and IVIPER perform when measuring the performance of a team of decision-tree agents?



Experiments

- MAVIPER and IVIPER policies utilize different features



Experiments

- MAVIPER teams are more robust

Team	MAVIPER	IVIPER	Imitation DT	Fitted Q-Iteration
Defender	**.77** (.01)	.33 (.01)	.24 (.03)	.004 (.00)
Adversary	**.42** (.03)	**.41** (.03)	**.42** (.03)	.07 (.01)

Outline

- Opportunities and Challenges in Applying Multi-Agent Reinforcement Learning
  - MARL for Security and Sustainability
  - Interpretable MARL
- Discussion and Summary

Summary

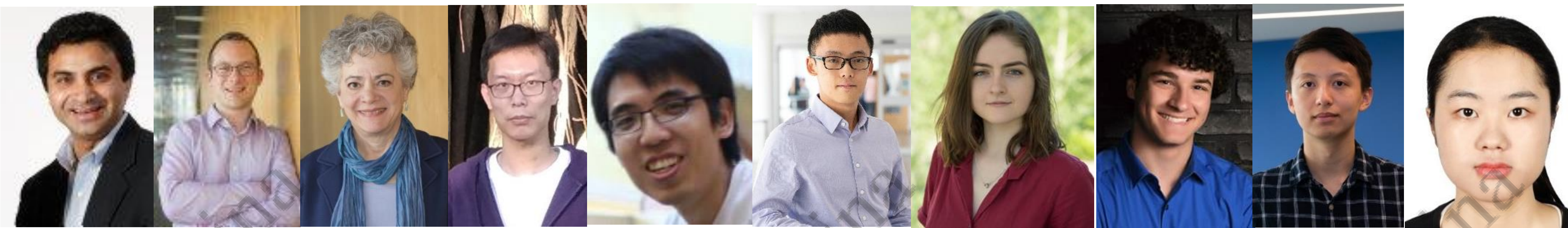
- MARL has great potential for real-world challenges in security and sustainability domains

- Fundamental challenges need to be addressed to fulfill the potential
  - Equilibrium refinement
  - Interpretability
  - … (sim2real gap, scalability, etc)

- Discussion: Other application domains? Other key challenges?

Acknowledgment

- Advisors, postdocs, students and all co-authors!



- Collaborators and partners
- Funding support

References

- [Lipton, 2018] Lipton, Zachary C. "The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery." Queue 16.3 (2018): 31-57.
- [Pyeatt, 2003] Pyeatt, Larry D. "Reinforcement learning with decision trees." 21 st IASTED International Multi-Conference on Applied Informatics. 2003.
- [Malias & Kudenko, 2015] Malialis, Kleanthis, and Daniel Kudenko. "Distributed response to network intrusions using multiagent reinforcement learning." Engineering Applications of Artificial Intelligence 41 (2015): 270-284.
- [Molnar, 2020] Molnar, Christoph. Interpretable machine learning. Lulu. com, 2020.
- [Milani & Topin, et al., 2022] Milani, Stephanie, Nicholay Topin, Manuela Veloso, and Fei Fang. "A survey of explainable reinforcement learning." arXiv preprint arXiv:2202.08434 (2022).
- [Bastani et al., 2018] Bastani, Osbert, Yewen Pu, and Armando Solar-Lezama. "Verifiable reinforcement learning via policy extraction." arXiv preprint arXiv:1805.08328 (2018).
- [Terry, 2021] Multi-Agent Deep Reinforcement Learning in 13 Lines of Code Using PettingZoo | by J K Terry | Towards Data Science