

Final Performance Report

Project Title: Explainable AI for Mission Planning and Execution with Interpretable Courses of Action

PI/Co-PI Names: Huy T. Tran* (PI), Girish Chowdhary*, R. S. Sreenivas*, Heather Filippini⁺

Institution Names: *University of Illinois at Urbana-Champaign, ⁺Illinois Applied Research Institute

Award No: N00014-20-1-2249

Period of Performance: 04/01/2020 - 12/31/2023



Contents

1	Abstract	2
2	Distribution Statement	3
3	Accomplishments	4
3.1	What were the major goals and objectives of the project?	4
3.2	What was accomplished towards achieving these goals?	4
3.2.1	RT #1: Classifying and interpreting decision sequences from AI systems as coherent COAs	6
3.2.2	RT #2: Discovering and incorporating commander biases into AI systems .	10
3.2.3	RT #3: Optimizing and Evaluating Hierarchical AI Systems	15
3.3	What opportunities for training and professional development did the project pro- vide?	26
3.4	How were the results disseminated to communities of interest?	27
3.5	What honors or awards were received under this project?	27
4	Technology Transfer	28
5	Participants	28
6	Students	28
7	Products	28

1 Abstract

Navy mission planning and execution requires the ability to consider a large number of potential courses of action (COAs) and ensure optimal decision-making in uncertain environments. However, current methods can only explore a limited number of COAs, do not accurately model commander biases, and struggle when facing intelligent adversaries. Recent advances in artificial intelligence (AI) have the potential to address these limitations, thus creating a paradigm shift in how the Navy operates. Yet several critical challenges limit our ability to fully realize this shift, many of which relate to a lack of explainability in deep learning-based AI systems and performance of explainable methods. *The overall goal of this project was to make progress towards this vision by developing methods for improved explainability, interpretation, and performance in AI systems for Navy mission planning and execution.* This project addressed these challenges through three primary research thrusts. Each thrust generated new methods and knowledge that can support the transition of AI systems to the Navy.

Research thrust one developed methods to identify coherent COAs from AI systems, including a heuristic temporal logic search algorithm and a neurosymbolic approach for learning more general temporal logic formulae from agent trajectories. *Research thrust two* developed methods to discover biases from decision sequences and incorporate biases into trained decision policies. To this end, we created a framework for an RL agent to adapt to biased teammates, methods for analyzing the impact of agent biases on team performance, and a survey of inverse single- and multi-agent reinforcement learning (RL) methods for discovering agent biases. *Research thrust three* developed methods to improve the performance of explainable RL agents, with a focus on hierarchical agents. Here, we created a method for training hierarchical RL agents that can handle complex adversaries, a multi-agent RL method for training large teams of coordinated agents in adversarial environments, a framework for minimizing communications in multi-agent RL, and a method for deploying spiking neural network-based RL policies on neuromorphic hardware for low SWaP applications.

Our results were disseminated to communities of interest through five manuscripts that were accepted or submitted to AI/robotics journals, conferences, and workshops, with another eight manuscripts currently in preparation for submission. The PIs also discussed results from the project through invited seminars. This project provided training and professional development for eight graduate students, three undergraduate students, one research engineer, and one post-doctoral scholar. Technology transfer was accomplished through follow-on funding from other DoD and civil agencies, hiring of students to DoD laboratories, and discussions with other DoD researchers.

2 Distribution Statement

DISTRIBUTION A. Approved for public release: distribution unlimited.

3 Accomplishments

3.1 What were the major goals and objectives of the project?

Effective mission planning requires the exploration of potential courses of action (COAs) that consider biases in force commanders; successful mission execution then requires the ability to react to unexpected situations that may occur during the mission. However, current planning processes can only explore a limited number of COAs and provide limited consideration of commander biases. Furthermore, while autonomy technologies offer new capabilities for mission execution, current systems are limited in highly dynamic and contested environments. Advances in artificial intelligence (AI), particularly within the field of deep reinforcement learning (RL), have the potential to address these limitations by generating COAs for mission planning and creating decision policies for executing plans. *The overall goal of this project was to make progress towards this vision by developing methods for improved explainability, interpretation, and performance in AI systems for Navy mission planning and execution.* More specifically, this project aimed to achieve the following technical objectives:

- develop methods to identify coherent Course of Actions (COAs) from AI systems;
- develop methods to discover and incorporate commander biases into AI systems;
- develop methods to optimize and evaluate hierarchical RL systems;
- assess developed methods using Navy-relevant simulations and hardware-based demonstrations.

The project was organized into the following three research thrusts (RTs) to achieve these objectives, with all three RTs addressing technical objective 4. Figure 1 provides an overview of the RTs and their relationships to each other.

- RT #1: Classifying and interpreting decision sequences from AI systems as coherent COAs;
- RT #2: Discovering and incorporating commander biases into AI systems;
- RT #3: Optimizing and evaluating hierarchical AI systems.

3.2 What was accomplished towards achieving these goals?

There were nine main technical accomplishments throughout the course of this project, summarized in the list below. Sections 3.2.1 to 3.2.3 discuss these accomplishments in more detail (technical approach and results), organized by research thrust.

1. **We created a heuristic temporal logic search algorithm for explaining RL agents.** This work was submitted as a manuscript titled “On Generating Explanations for Reinforcement Learning Policies: An Empirical Study” to the 2024 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* [1].

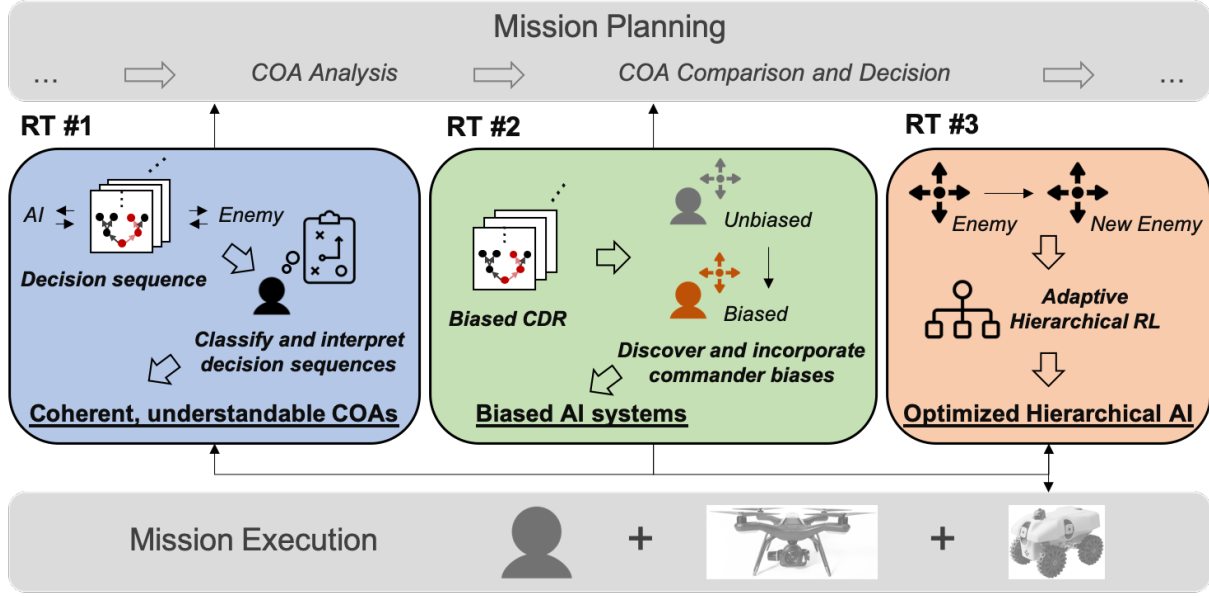


Figure 1: Overview of the project research thrusts (RTs). Our project was split into three main RTs and aims to support Navy mission planning and execution.

2. **We created a neurosymbolic approach for learning general temporal logic formulae to explain AI agents.** This work is ongoing with a manuscript in preparation.
3. **We created a framework for RL Agents to adapt to biased teammates.** This work was presented as a manuscript titled “Coordination in Ad Hoc Teams with Generalized Policy Improvement” at the *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop on Advances in Multi-Agent Learning - Coordination, Perception, and Control* [2]. A follow-on manuscript is in preparation for submission to *IEEE Robotics and Automation Letters (RA-L)*.
4. **We created methods for analyzing the impact of agent biases on team performance.** This work is ongoing with a manuscript in preparation.
5. **We are completing a survey of inverse single- and multi-agent RL methods for discovering agent biases.** This work is ongoing with a manuscript in preparation.
6. **We created a hierarchical RL algorithm that can handle complex adversaries.** This work was presented as a manuscript titled “Feature Specialization and Clustering Improves Hierarchical Sub-task Learning” at the *Adaptive and Learning Agents (ALA) Workshop at AAMAS 2022* [3]. A follow-on manuscript is in preparation for submission to the *Journal of Autonomous Agents and Multi-agent Systems (JAAMAS)*.
7. **We created a MARL algorithm for coordinating and explaining large teams of MARL agents in adversarial environments.** This work was published and presented as a manuscript titled “Disentangling Successor Features for Coordination in Multi-agent Reinforcement Learning” at the *21st International Conference on Autonomous Agents and Multiagent Sys-*

tems (AAMAS 2022) [4]. A follow-on manuscript is in preparation for submission to the *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*.

8. **We created a framework for minimizing communications in MARL.** This work was submitted as a manuscript titled “Coordination Machines for Minimizing Communication in Multi-Agent Reinforcement Learning” to the *2024 Reinforcement Learning Conference (RLC) Workshop on Coordination and Cooperation in Multi-Agent Reinforcement Learning (CoCoMARL)*, with a follow-on manuscript in preparation for submission to *IEEE Robotics and Automation Letters (RA-L)*.
9. **We created a method for deploying SNN-based RL policies on neuromorphic hardware for low SWaP applications.** This work is ongoing with a manuscript titled “SPAC-Net: Spiking Actor-Critic Network” in preparation for submission to *IEEE Robotics and Automation Letters (RA-L)*.

3.2.1 RT #1: Classifying and interpreting decision sequences from AI systems as coherent COAs

The objective of this thrust was to develop methods to identify coherent COAs from AI systems, with a focus on classifying and interpreting decision sequences from RL agents. To this end, we first created a new heuristic search algorithm for generating a cogent explanation for a given target RL policy using temporal logic [1]. We then created a neurosymbolic method for learning the temporal logic formula that best explains sampled trajectories from an AI agent, which considers a more general class of formulae that considered in prior work. We discuss these two accomplishments below.

Accomplishment 1: A Heuristic Temporal Logic Search Algorithm for Explaining RL Agents.

We first created a heuristic tree-search algorithm for identifying the linear temporal logic (LTL) formula that best explains a given RL policy. This work was submitted as a manuscript titled “On Generating Explanations for Reinforcement Learning Policies: An Empirical Study” to the *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* [1].

Our method aims to identify an explanation that defines both the ultimate objectives achieved by a policy, as well as the operational constraints/conditions maintained throughout execution. For example, an explanation for an RL agent trained to win a game of capture-the-flag (CTF) could be that it “aims to eventually capture the red team’s flag while always avoiding red/adversary agents.” Our approach centers on a family of LTL formulae that is endowed with a concept of neighborhood, which in turn is amenable to a local-search algorithm. In our method, each potential LTL-explanation is effectively translated into an RL policy using established techniques in the literature [5]. We then gauge the alignment of this representative policy with the target policy by calculating a weighted average of KL (wKL) divergence values comparing action distributions from the representative and target policies over randomly sampled states. Should a neighboring LTL-explanation exhibit superior alignment with the target policy according to this metric, it supplants the current explanation. This iterative process persists until none of the neighboring LTL-explanations surpasses the present one, thus establishing a local optimum as the recommended LTL-explanation. To enhance robustness, we propose additional neighborhood expansion and extension heuristics,

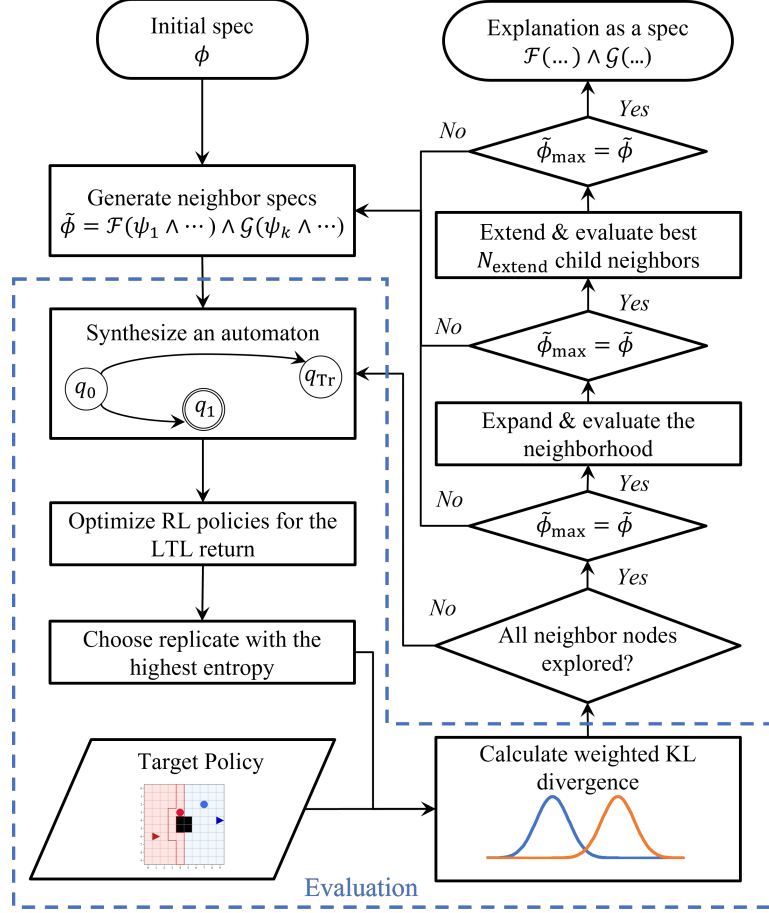
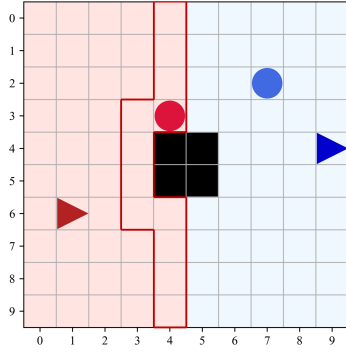


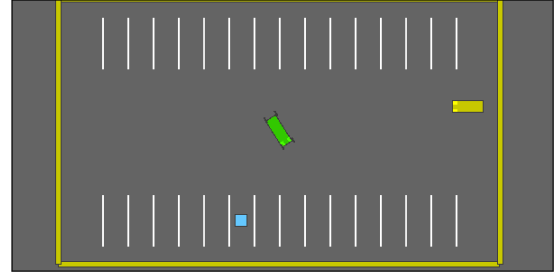
Figure 2: Overview of our approach for generating a human-readable explanation of a given RL policy. Our method includes steps for generating an RL policy to satisfy a potential LTL explanation, and then comparing that policy to the given target policy.

as well as a multi-start implementation of the search that generates the top- k (e.g., $k = 10$) LTL specifications as candidate explanations.

We demonstrated our algorithm in two simulated environments: (1) a 1 vs. 1 CTF game (shown in Figure 17) and (2) a car-parking environment (shown in Figure 3). Table 1 shows results from the CTF environment, where we defined the target policy for the blue agent as an RL policy optimized for the LTL-specification $\mathcal{F}(\psi_{BA,RF} \wedge \neg\psi_{RA,BF}) \wedge \mathcal{G}(\neg\psi_{BA,RA} \vee \psi_{BA,BT})$. In plain English, this specification requires that: “*Eventually, the blue agent reaches the red flag and does not reach the blue flag. Globally, the blue agent does not encounter the red agent or it stays in the blue territory.*” We saw that Searches 1 and 5 successfully found the true target specification as the specification with the highest utility (i.e., lowest wKL divergence) among searched specifications. Search 1 searched 8.13% of the search space while Search 5 searched 10.2% of it. The second best search result is Search 6, whose explanation is: “*Eventually, the blue agent captures the red flag. Globally, the blue agent is not inside the blue territory while the red agent does not capture the blue flag, or the blue agent does not encounter the red agent.*” This explanation is close to the target LTL specification and consistent with the game dynamics. These results suggest that our search algorithm can efficiently find an exact match for the target policy’s explanation.



(a)



(b)

Figure 3: The environments used for testing our LTL search algorithm. (a) Visualization of our 1 vs. 1 CTF game. Black squares are obstacles, triangles are flags, and circles are agents. The region highlighted by solid red lines is the border region for the red agent. (b) Visualization of the parking environment. The green rectangle is the ego vehicle, the blue square is the landmark, and the yellow rectangle is the other vehicle. Yellow walls mark the boundary.

Table 1: CTF experiment search results. The target policy was successfully found in Searches 1 and 5.

Search	LTL specification	wKL div. [-]	Searched specs [%]
1	$\mathcal{F}(\psi_{BA,RF} \wedge \neg\psi_{RA,BF}) \wedge \mathcal{G}(\neg\psi_{BA,RA} \vee \psi_{BA,BT})$	8.00×10^{-8}	8.13
2	$\mathcal{F}((\neg\psi_{RA,BF}) \vee (\psi_{BA,RF})) \wedge \mathcal{G}(\neg\psi_{BA,RA} \vee \psi_{BA,BT})$	5.95×10^{-6}	9.06
3	$\mathcal{F}((\neg\psi_{BA,RA}) \vee (\neg\psi_{RA,BF})) \wedge \mathcal{G}((\psi_{BA,BT}) \vee (\psi_{BA,RF}))$	8.52×10^{-5}	6.56
4	$\mathcal{F}(\neg\psi_{BA,BT} \wedge \neg\psi_{RA,BF}) \wedge \mathcal{G}((\neg\psi_{BA,RF}) \vee (\neg\psi_{BA,RA}))$	1.57×10^{-6}	6.56
5	$\mathcal{F}(\psi_{BA,RF} \wedge \neg\psi_{RA,BF}) \wedge \mathcal{G}(\neg\psi_{BA,RA} \vee \psi_{BA,BT})$	8.00×10^{-8}	10.2
6	$\mathcal{F}(\psi_{BA,RF}) \wedge \mathcal{G}((\neg\psi_{BA,BT} \wedge \neg\psi_{RA,BF}) \vee (\neg\psi_{BA,RA}))$	7.42×10^{-7}	8.44
7	$\mathcal{F}(\psi_{BA,RA}) \wedge \mathcal{G}((\neg\psi_{BA,BT}) \vee (\neg\psi_{BA,RF} \wedge \neg\psi_{RA,BF}))$	1.18×10^{-5}	8.59
8	$\mathcal{F}(\neg\psi_{BA,BT}) \wedge \mathcal{G}((\neg\psi_{BA,RA} \wedge \psi_{BA,RF}) \vee (\neg\psi_{RA,BF}))$	1.46×10^{-8}	6.56
9	$\mathcal{F}((\psi_{BA,RA}) \wedge (\neg\psi_{BA,RF})) \wedge \mathcal{G}((\psi_{BA,BT}) \vee (\neg\psi_{RA,BF}))$	1.18×10^{-5}	11.4
10	$\mathcal{F}((\psi_{RA,BF}) \vee (\neg\psi_{BA,BT})) \wedge \mathcal{G}((\neg\psi_{BA,RA}) \vee (\psi_{BA,RF}))$	1.62×10^{-5}	7.19

2gray!25white

Figure 4 shows a partial trace of the search tree produced by Search 5. We see that the extension step was used in the third step of the search, which successfully kept the search from getting stuck in a sub-optimal local minimum early in the search process. More specifically, when the parent node in the third step was the local minimum (wKL divergence of 9.95×10^{-6}) among its neighborhood after an expansion step, the red and green nodes’ neighborhoods were opened first and second, respectively, by the extension step. Subsequently, the neighborhood of the blue node next to the green node was opened and found a new local minimum whose wKL divergence value was 8.00×10^{-8} .

Table 2 shows results from the parking environment, where we defined the target policy of the ego vehicle as an RL policy optimized for specification $\mathcal{F}(\psi_{ego,goal}) \wedge \mathcal{G}(\neg\psi_{ego,other} \wedge \neg\psi_{ego,wall})$, which requires that: “*Eventually, the ego vehicle reaches the goal. Globally, the ego vehicle does not hit the other vehicle and walls.*”. We saw that Searches 1, 5, and 7 successfully found the true target specification, with each searching 37.5%, 29.2%, and 33.3% of the search space, re-

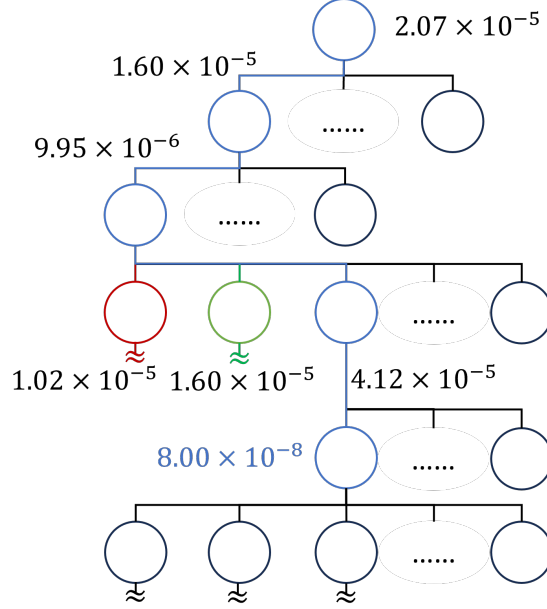


Figure 4: A partial trace of Search 5 from Table 1. Circles represent potential LTL explanations for the target policy, with their corresponding wKL divergence values shown. The blue nodes and lines show how the local minimum moved as the search proceeded.

Table 2: Parking experiment results. The target policy was successfully found in Searches 1, 5, and 7.

Search	LTL specification	wKL div. [-]	Searched specs [%]
1	$\mathcal{F}(\psi_{\text{ego}, \text{goal}}) \wedge \mathcal{G}(\neg\psi_{\text{ego}, \text{other}} \wedge \neg\psi_{\text{ego}, \text{wall}})$	0.00	37.5
2	$\mathcal{F}(\neg\psi_{\text{ego}, \text{other}} \wedge \psi_{\text{ego}, \text{wall}}) \wedge \mathcal{G}(\neg\psi_{\text{ego}, \text{goal}})$	7.35×10^{-4}	28.1
3	$\mathcal{F}(\neg\psi_{\text{ego}, \text{other}} \wedge \psi_{\text{ego}, \text{wall}}) \wedge \mathcal{G}(\neg\psi_{\text{ego}, \text{goal}})$	7.35×10^{-4}	27.1
4	$\mathcal{F}(\psi_{\text{ego}, \text{wall}}) \wedge \mathcal{G}(\neg\psi_{\text{ego}, \text{goal}} \vee \psi_{\text{ego}, \text{other}})$	6.40×10^{-4}	30.2
5	$\mathcal{F}(\psi_{\text{ego}, \text{goal}}) \wedge \mathcal{G}(\neg\psi_{\text{ego}, \text{other}} \wedge \neg\psi_{\text{ego}, \text{wall}})$	0.00	29.2
6	$\mathcal{F}(\psi_{\text{ego}, \text{other}}) \wedge \mathcal{G}(\neg\psi_{\text{ego}, \text{goal}} \vee \neg\psi_{\text{ego}, \text{wall}})$	4.61×10^{-4}	44.8
7	$\mathcal{F}(\psi_{\text{ego}, \text{goal}}) \wedge \mathcal{G}(\neg\psi_{\text{ego}, \text{other}} \wedge \neg\psi_{\text{ego}, \text{wall}})$	0.00	37.5
8	$\mathcal{F}(\psi_{\text{ego}, \text{other}}) \wedge \mathcal{G}(\neg\psi_{\text{ego}, \text{goal}} \vee \neg\psi_{\text{ego}, \text{wall}})$	4.61×10^{-4}	33.3

spectively. The second best results were Searches 6 and 8, which found the same solution whose explanation is: “*Eventually, the ego vehicle encounters the other agent. Globally, the ego vehicle does not reach the goal or does not hit the wall.*” This explanation captures a failure case, where the ego vehicle sometimes does not reach the goal because it collides with the other vehicle, though it does successfully avoid walls. This result suggests that our search can find the true target explanation, but can also capture unlikely, but observed, behaviors from the target policy that occur due to the stochastic nature of the environment.

2gray!25white

Accomplishment 2: A Neurosymbolic Approach for Learning General Temporal Logic Formulae to Explain AI Agents. Two limitations of our search algorithm are that it is heuristic and limited to a small, restricted class of LTL formulae. We therefore then explored neurosymbolic methods for efficiently searching through a more general class of formulae to explain a given AI

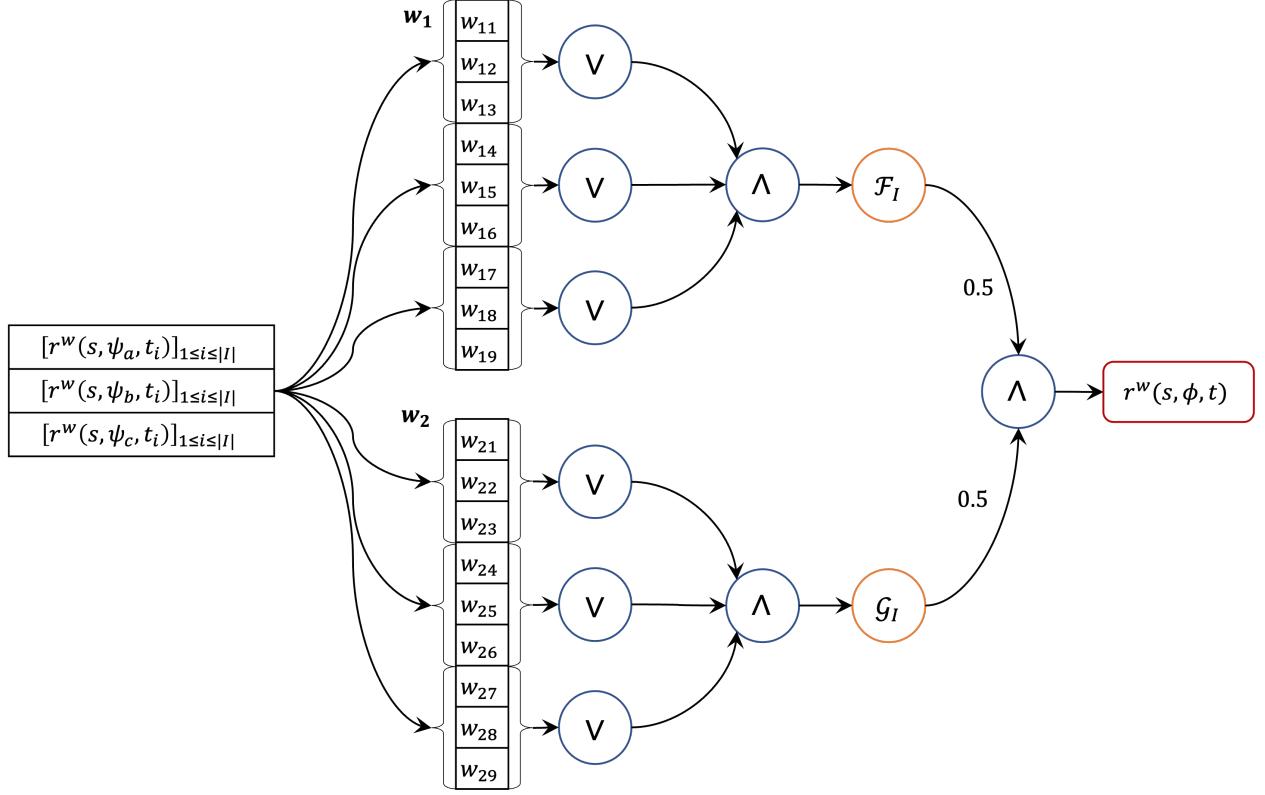


Figure 5: Example architecture of our proposed neural network for inferring a wSTL formula explanation of an AI agent. In this example, there are three input atomic predicates (ψ_a, ψ_b , and ψ_c), w_1 and w_2 are the optimized weight vectors, and ϕ is the inferred explanation for the given trajectory.

agent. This work is ongoing with a manuscript in preparation.

Our key idea is to use the syntax of weighted signal temporal logic (wSTL) to generate a formula that best explains the objective of an AI agent, given sampled trajectories of that agent. wSTL extends LTL by incorporating weights that allow one to model objectives with varying importance given to each part of the objective. For example, returning to the CTF game, a possible wSTL objective for an agent could be: “*Eventually, the blue agent reaches the red flag with 80% importance and kills the red agent with 20% importance. Globally, the red agent does not reach the blue flag with 100% importance.*” The inclusion of continuous weights within wSTL also allows one to leverage backpropagation within neural networks to efficiently search for the weights that best predicts given trajectory data. Our preliminary results show that our approach can find a formula that is a reasonably close match to the target formula. More specifically, given a target formula of $0.5\mathcal{F}(a) \wedge 0.5\mathcal{G}(0.5b \wedge 0.5c)$, our method inferred a formula of $0.5\mathcal{F}(a) \wedge 0.5\mathcal{G}(0.695a \vee (0.152b \wedge 0.253c))$, which is a slightly looser (but still correct) form of the desired target formula.

3.2.2 RT #2: Discovering and incorporating commander biases into AI systems

The objective of this thrust was to develop methods to discover biases from decision sequences and incorporate biases into trained decision policies. To this end, we first created a new method for an RL agent to adapt to a new teammate (i.e., one not seen during training) that may have a different

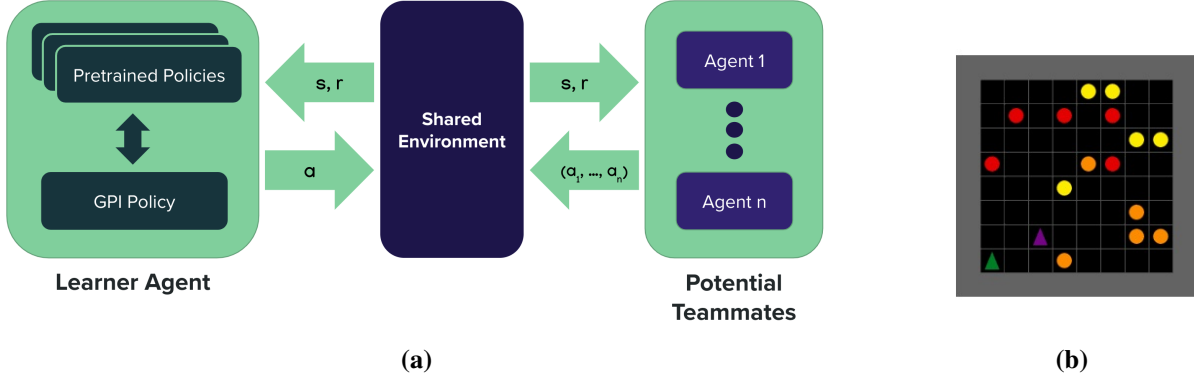


Figure 6: Our general framework for adapting to biased teammates and the environment used for testing. (a) The key idea of our GSAT+ method for adapting to teammates with different biases is to leverage a pretrained library of policies in a principled manner that offers theoretical guarantees and performance bounds under certain assumptions. (b) Our multi-agent collect game environment. The two agents are represented by triangles, where green is the learner and purple is the uncontrolled agent. Circles represent objects and different object types have different colours..

bias (or objective) than expected. We then began to explore game theoretic methods for analyzing and understanding the impact of agent biases on team performance. Finally, we began to explore inverse RL (IRL) methods for discovering agent biases. We discuss these three accomplishments below.

Accomplishment 3: RL Agents that can Adapt to Biased Teammates. We first created a framework for an RL agent (called the “learner agent”) to adapt to a new teammate, where that teammate optimizes a different/biased objective [2]. This work was presented as a manuscript titled “Coordination in Ad Hoc Teams with Generalized Policy Improvement” at the *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop on Advances in Multi- Agent Learning - Coordination, Perception, and Control* [2]. A follow-on manuscript is in preparation for submission to *IEEE Robotics and Automation Letters (RA-L)*.

As an example of our problem set up, a learner agent might have been trained to search for the flag in a game of CTF, while a new teammate might have instead been trained to engage the enemy — this biased objective of the new teammate could lead to poor coordination and team performance if the learner does not effectively adapt to the teammate. Our approach to addressing this problem uses the RL framework of generalized policy improvement (GPI) with successor features (SFs) [6], and is named GSAT+ (GPI with SFs for Ad Hoc Teaming). A GPI policy is a policy that operates over a library of pretrained policies and is guaranteed to be better than each policy within the library. SFs are a useful model for transferring each policy within the library to a new task — prior work focuses on using GPI and SFs to enable a single agent to adapt to a new reward, like collecting different colored coins. Our GSAT+ method extends this prior work to a multi-agent setting by enabling adaptation to new teammates with different biases. The basic idea of GSAT+ is thus to pretrain a library of policies for different potential teammates, after which the learner agent uses a GPI policy to effectively use that library to adapt to a new teammate. Our framework is summarized in Figure 6a.

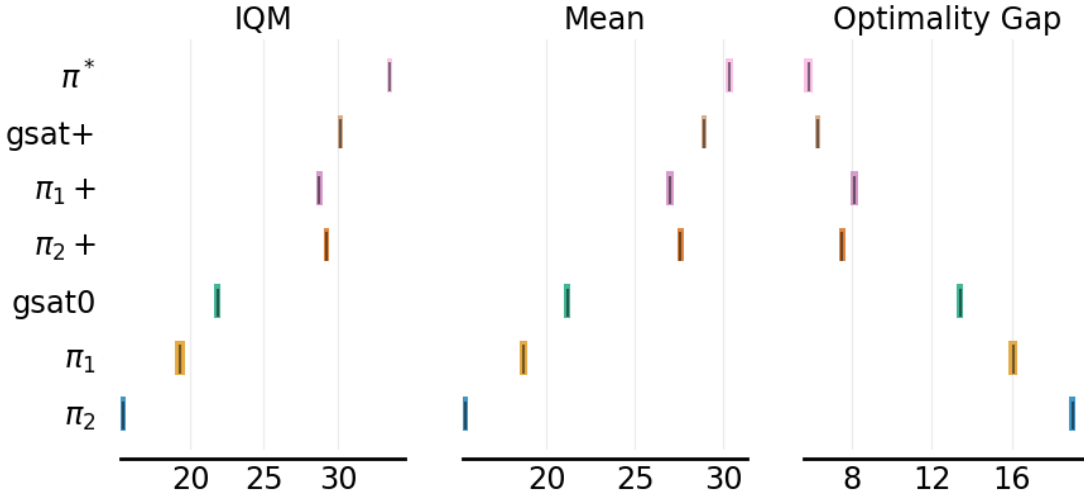


Figure 7: Comparing performance (total returns) for our method (gsat+), a zero-shot variant of our method (gsat0), and baselines. We see that our method outperforms all baselines, except for π^* , which represents the oracle solution (i.e., the policy allowed to fully pretrain with the new teammate). Higher mean, higher IQM scores, and lower optimality gaps are better, with 95% confidence intervals (CIs) shown by colored regions. The CIs are estimated using the percentile bootstrap with stratified sampling with 50,000 bootstrap resamples. The optimality gap is calculated with respect to the IQM of π^* . We ran 3 replicates for each algorithm.

We demonstrated our method in a simulated “collect” game where two agents work together to collect coins of different colors, shown in Figure 6b. Figures 7 and 8 show results comparing our method to baselines. For these results, our learner agent was given a library containing two pretrained policies and partnered with a new teammate trained for a reward different from those seen in the library. The learner agent therefore had to learn how to adapt to this new teammate with a bias not seen during pretraining. Our baselines included each of the pretrained policies within the library, as well as an agent trained from scratch with the new teammate. Our choice of baselines was inspired by state-of-the-art ad hoc teaming methods, such as PLASTIC-policy [?]. When confronted with new teammates, these methods use the pretrained policy in a library that is best suited for the new team, typically based on inferred similarity to the new teammate. Our results do not include this inference process and simply show the performance obtained by each pretrained library policy. We included an agent trained from scratch to represent the ideal solution — that is, an agent allowed to train as long as needed with the new teammate. Our results showed that for this setting, our method showed better performance than all baselines other than the ideal oracle solution (π^*), suggesting our framework is a promising approach for adapting to new teammate biases.

We also derived a new theorem that provides an upper bound on the suboptimality of our method, which is summarized in Theorem 1, with details and definitions provided in [2]. More specifically, we bound the difference between the action-value function of a deterministic policy for the learner optimized for a new teammate and executed with that new teammate, and the action-value function of a deterministic policy for the learner optimized for a prior teammate but executed with the new teammate. This theorem extends Theorem 2 from [6] to consider adaption to new

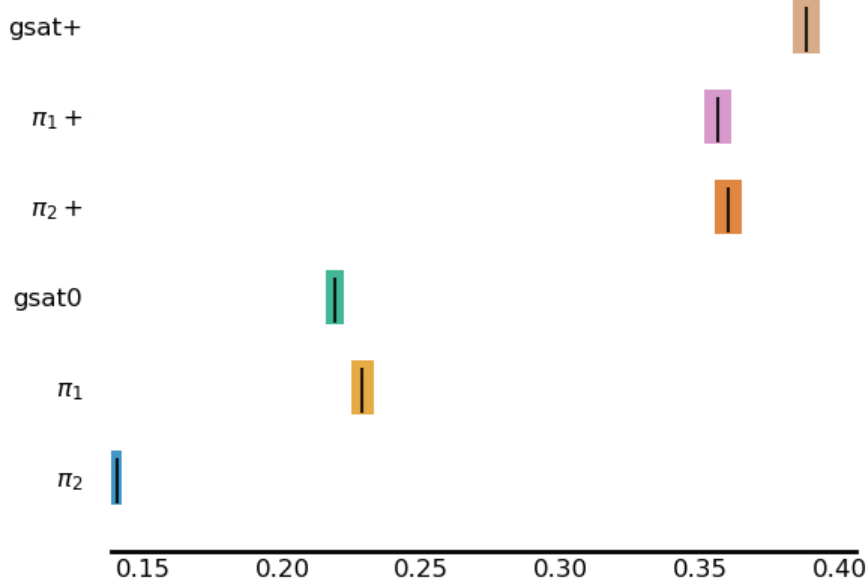


Figure 8: Comparing the probability of improvement of a given algorithm compared to the oracle solution (π^*). We again see that our method (gsat+) outperforms all considered baselines. The interval estimates are based on stratified bootstrap with independent sampling with 2000 bootstrap resamples.

teammates — prior work only considered adaptation of a single agent to new rewards.

Theorem 1. Suppose the agent has optimized policies in n different ad hoc MMDPs within $\mathcal{M}^\phi \setminus M_k$, so that we have n policies $\pi_1^{a*}, \dots, \pi_n^{a*}$. First, we approximate the action-value function of these policies when executed in M_k such that,

$$|Q_k^{j*}(s, a^a) - \tilde{Q}_k^{j*}(s, a^a)| \leq \epsilon \quad (1)$$

for all $s \in \mathcal{S}$, $a^a \in \mathcal{A}$, where ϵ is the approximation error and $j \in \{1, \dots, n\}$.

Now, given the set $\{\tilde{Q}_k^{1*}, \tilde{Q}_k^{2*}, \dots, \tilde{Q}_k^{n*}\}$, we define the learner policy π^a as,

$$\pi^a(s) \in \operatorname{argmax}_{a^a} \max_j \tilde{Q}_k^{j*}(s, a^a). \quad (2)$$

with its corresponding action-value function denoted by $Q_k^{\pi^a}$. Then we can bound its performance as,

$$|Q_k^*(s, a^a) - Q_k^{\pi^a}(s, a^a)| \leq \frac{2}{1 - \gamma} (\epsilon_r + 2\gamma d_{TV} \cdot \Omega + \epsilon) \quad (3)$$

where we define $\epsilon_r := \max_{s, a^a} |r_k^a(s, a^a) - r_j^a(s, a^a)|$, $d_{TV}(\tilde{p}_k, \tilde{p}_j) := \frac{1}{2} \max_{s, a^a} \sum_{s'} |\tilde{p}_k(s'|s, a^a) - \tilde{p}_j(s'|s, a^a)|$, and $\Omega := \max_{s, a^a} Q_j^* + \frac{1}{2}$. Here, $j = \operatorname{argmax}_j \tilde{Q}_k^{j*}$.

Accomplishment 4: Methods for Analyzing the Impact of Agent Biases on Team Performance. While our GSAT+ method enables an agent to adapt to a new biased teammate, it provides limited understanding of how much a teammate’s (or other agent’s) biases affect the overall

team (or system) objective. We therefore created a method to quantify the impact of different agent biases on the overall team objective, in a manner that would allow a mission planner to determine when additional intervention may be required to achieve satisfactory overall team performance. This work is ongoing with a manuscript titled “Predicting social outcomes in collective autonomous air mobility systems with self-interested agents” in preparation for submission to the *2025 AIAA SciTech Form*, with a follow-on submission to the *AIAA Journal of Aerospace Information Systems*.

Our approach extends the classical idea of Price of Anarchy (PoA) to enable its evaluation as a function of any given state that may be observed by the system of interest — by creating a state-dependent PoA metric, we can now evaluate the PoA given any arbitrary state to determine if, for example, a team of biased agents (i.e., agents with different objectives) will perform well-enough with respect to the overall team objective given the current scenario. This scenario could be defined by the environmental conditions and adversary disposition, among other things. If the estimated PoA metric is not satisfactory, then a mission planner could intervene, for example, by sending in additional forces or changing the mission plan. Our idea for calculating such a PoA metric is simple — we simply calculate it as a ratio of the state-value function for the joint policy of the unbiased agents (i.e., agents optimizing the team objective) with respect to the overall team objective, to the state-value function for the joint policy of the biased agents (i.e., agents optimizing their own biased objectives) with respect to the overall team objective. This calculation is shown in Equation (4),

$$PoA := \frac{V_{MMDP}^{\pi_{MMDP}^*}(s)}{V_{MMDP}^{\pi_{MG}^*}(s)}, \quad (4)$$

where π_{MMDP}^* is the joint policy of the unbiased agents (optimized using a multi-agent Markov decision process (MMDP) model), π_{MG}^* is the joint policy of the biased agents (optimized using a Markov game (MG) model), and $V_{MMDP}^{\pi}(s)$ is the state-value function of a policy π with respect to the overall team objective.

We are in the process of demonstrating our method in a simulated drone firefighting scenario, where two agents aim to put out a wildfire and prevent it from spreading throughout a region. Inspired by [7], we model agent biases as preferences for preventing the fire from spreading to different regions — that is, each agent aims to protect all areas, but most strongly prefers to protect its own area. Our goal is to generate a map of the environment that shows how our PoA metric varies as a function of the initial location of a fire. A preliminary version of such a map is shown in Section 3.2.2, where larger values on the color bar indicate initial fire locations where the self-interested nature of these drones is resulting in many trees burning across the entire map.

Accomplishment 5: A survey of inverse single- and multi-agent RL methods for discovering agent biases. Finally, we are in the process of creating a survey of current state-of-the-art IRL methods that could be used to discover agent biases from data, with a focus on multi-agent IRL. This work is ongoing with an extended abstract titled “A Review of Inverse Reinforcement Learning for Aviation” in preparation for submission to the *2025 AIAA SciTech Form*, with a follow-on submission to the *AIAA Journal of Aerospace Information Systems*.

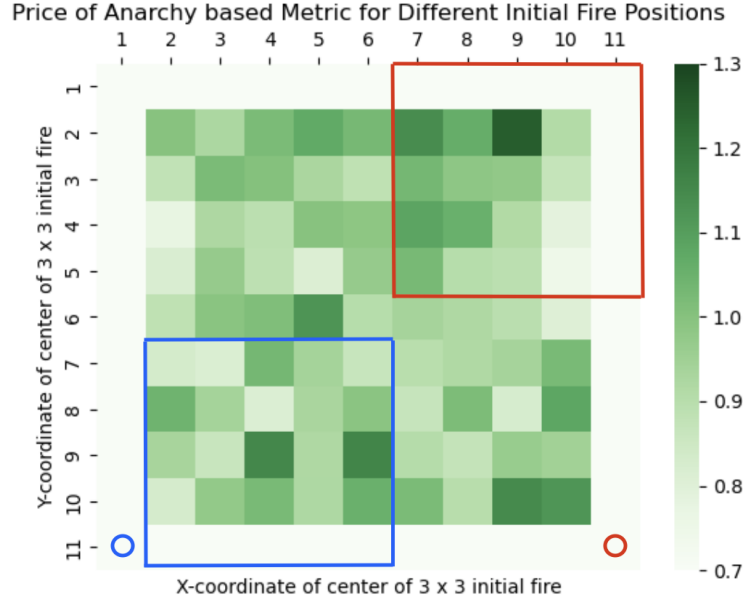


Figure 9: A preliminary PoA heat map for a drone firefighting application. The blue drone (blue circle) is biased towards protecting its own blue area from the fire, while the red drone (red circle) similarly is biased towards protecting its own red area. We aim to generate heat maps that help us understand which initial fire locations will be negatively affected by the biased nature of these drones. Similar methods could be used to analyze when biases within military forces will negatively affect overall team performance and require external interventions by higher-up commanders.

3.2.3 RT #3: Optimizing and Evaluating Hierarchical AI Systems

The objective of this thrust was to develop methods to improve the performance of explainable RL agents, with a focus on hierarchical agents. To this end, we first created a new method for training hierarchical RL agents that can handle complex adversaries. We then created a new multi-agent RL (MARL) method for coordinating teams of agents in adversarial environments, with a focus on explaining their behaviors and scaling to large teams (30 or more agents). We also created a new framework for minimizing communications within a team of agents while ensuring the team satisfies a given mission success rate. Finally, we began to explore spiking neural network (SNN)-based methods for training RL policies to support their deployment on low size, weight, and power (SWaP) systems. We discuss these four accomplishments below.

Accomplishment 6: Hierarchical RL Agents that can handle Complex Adversaries. We created a new neuroinspired algorithm, SNAC (Specialized Neurons And Clustering), for creating hierarchical RL agents that can handle dynamic and intelligent adversaries. This work was presented as a manuscript titled “Feature Specialization and Clustering Improves Hierarchical Sub-task Learning” at the *Adaptive and Learning Agents (ALA) Workshop at AAMAS 2022* [3]. A follow-on manuscript is in preparation for submission to the *Journal of Autonomous Agents and Multi-agent Systems (JAAMAS)*.

There is strong evidence that humans navigate in a hierarchical manner which leverages several types of specialized neurons which encode vectorial (distance and direction) information between

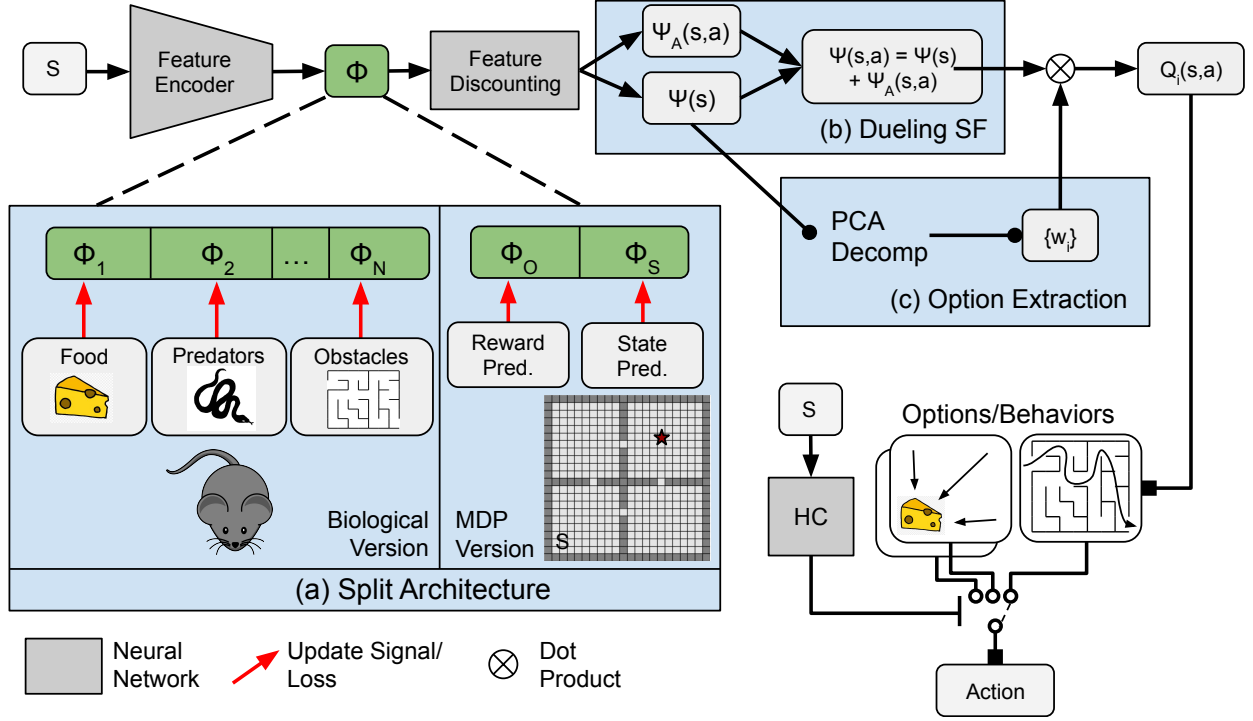


Figure 10: The overall architecture of our SNAC algorithm for hierarchical RL in complex adversarial environments. (a) A neural network feature encoder learns a split latent space, ϕ , broken into N distinct parts, each encoding a unique aspect of the environment. We use state and reward prediction for the split within an MDP model. (b) The SFs, ψ , are learned with a dueling architecture. (c) The SF representation ψ is decomposed with PCA, creating a set of principle components that are combined with SFs to create options. (d) A generic hierarchical RL framework, where learned options are used to solve an overall task.

goal locations. For example, place cells, object cells, and head direction cells are known to fire at specific locations, near specific objects, and in head directions, respectively, within the mammalian hippocampus. Furthermore, in the mouse entorhinal cortex, a feature space defined by clustering such specialized cells appears to form a topological map used for spatial cognition. Inspired by these ideas, we aimed to answer the following question: “*Can feature specialization and clustering of a topological map enable hierarchical RL agents to decompose environments into a task subspace, and will such decomposition result in significantly improved learning speed and performance?*” To this end, we first created new methods to model dynamic elements in the environment (like adversaries) and subsequently generate options (i.e., low-level policies) that specifically interact with those elements. Our key idea is to mimic the task-specialization of the entorhinal cortex by splitting a latent SF representation into specialized components, each of which encodes a unique element of the environment, such as spatial location or adversary information. We then decompose each component to generate a set of specialized options that relate to those unique elements, in a manner that does not require any domain-specific knowledge. We also created new methods to recover a complete task subspace of an environment from its SF representation — existing work focuses on identifying unique states, most commonly bottlenecks, to be used as option sub-goals, which cannot sufficiently model more complex tasks that require a set of options which completely

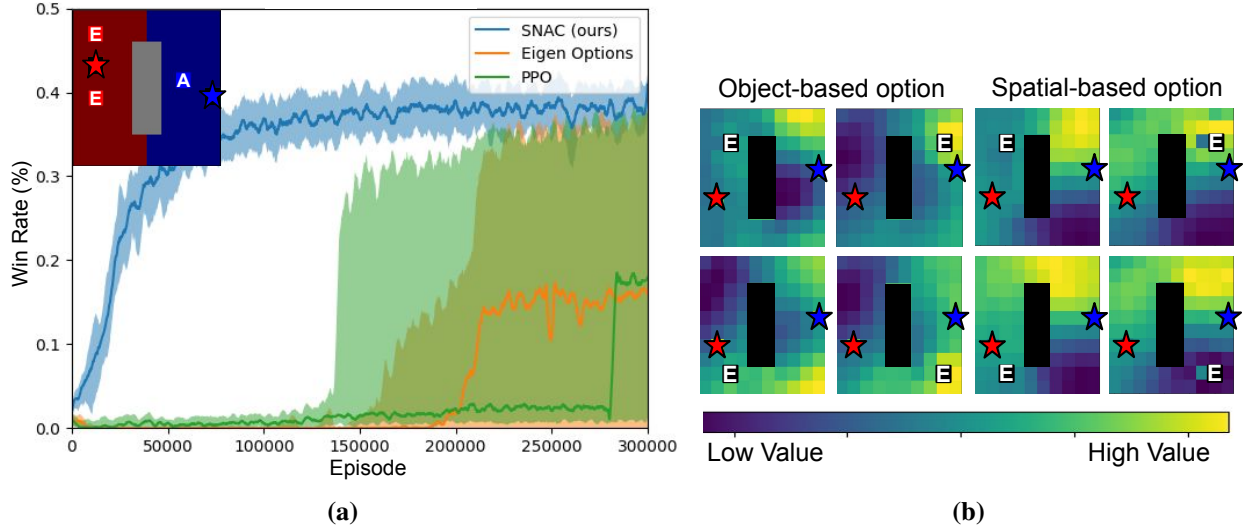


Figure 11: (a) Comparing performance during training in a 1v2 Adversarial CTF game. Our method (SNAC) significantly outperforms a state-of-the-art baseline (Eigen Options) and a standard PPO implementation, likely due to its ability to model dynamic adversaries. One controlled agent competes against two adversarial enemies to capture their flag while defending its own. Bold line is the median, and shaded region is the 25-75 percentile [10 Replicates]. **(b) Resultant options from SNAC.** The goal flags are marked with stars and the enemy position is marked by an “E”. The coloring represents the relative value of the position, where yellow indicates more value. Spatial-based options navigate to specific spatial locations, agnostic of the enemy position. Object-based options generated by SNAC are highly reactive to the enemy position; this specific option is highly defensive and moves to position the agent between the enemy and its own flag.

spans the environment. Our key idea is to mimic the brain’s use of a cluster subspace by using principle component analysis (PCA) to compress an SF topological map into a basis for k -means clustering. We use this subspace to generate options that allow agents to navigate to key clusters in the environment and perform micro-level control within those clusters. Our overall architecture is summarized in Section 3.2.3.

We demonstrated our method in two simulated environments: (1) a 1 vs. 2 CTF game and (2) the Alien game from Atari. Figure 11a shows results for our method (SNAC) and two baselines — we see that SNAC significantly outperforms both baselines, likely due to its ability to better model dynamic adversaries. Section 3.2.3 shows a visualization of high value states for different options learned by SNAC. We see that the “Object-based option” is more reactive to the location of the enemy (shown by “E”) than the “Spatial-based option”, again due to our method’s emphasis on modeling dynamic objects in the environment. Figure 12 shows results for SNAC in the Atari game Alien. Here, we again see that SNAC outperforms our baselines in this environment. However, we see that SNAC (and baselines) appear to plateau in performance near the end of training. This plateauing demonstrates the primary limitation of current SF methods: they perform well on tightly constrained environments, like grid worlds, where it is reasonable to explore a majority of the states during the preliminary training of the SF representation. However in larger environments, such as this one, the exploration of states distant from starting locations is limited, leading to an inaccurate SF representation ψ and options that struggle to generalize to the rest of the environment. We

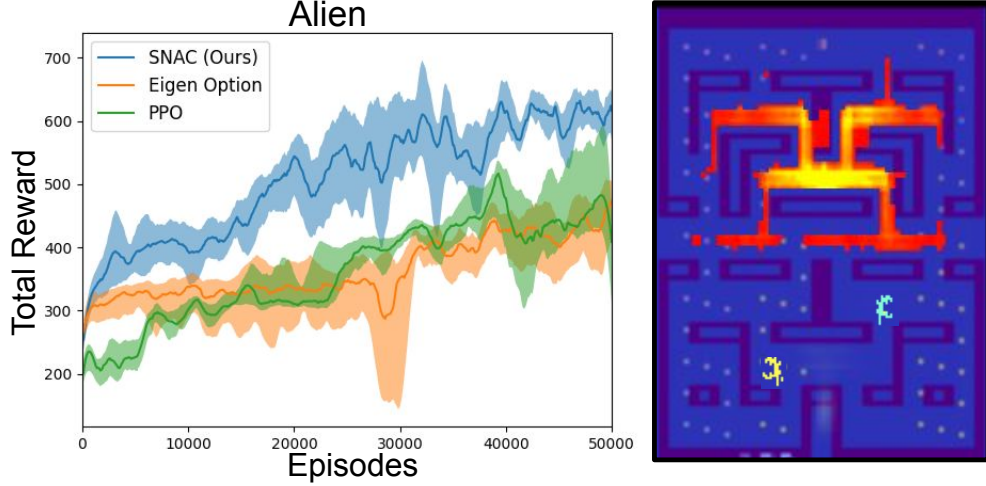


Figure 12: Comparing Performance in the Atari game Alien. Both option generation methods significantly outperform traditional RL in the first 5000 episodes. After this point, Eigenoptions performance suffers and drops to that of PPO, while our SNAC method continues to outperform both baselines. However, there is a performance plateau of SNAC near the end of training, likely due to limited exploration during training of the underlying SF representation, caused by the random walk exploration method typically used to maintain a reward-unbiased topological map. State visitation during SF training is highlighted on the right and shows less than half the map is visited.

highlight the states explored during SF training in the right figure of Figure 12, which shows that over half of the environment is never explored during training. We expect that implementations which expansively explore large action spaces will improve performance, and leave this direction for future work.

Accomplishment 7: Coordinating and Explaining Large Teams of MARL Agents in Adversarial Environments. We created a new algorithm, DISSC (Disentangled SFs for Coordination), for training large teams of agents to coordinate while facing complex adversaries, with an emphasis on explaining their behaviors. This work was published and presented as a manuscript titled “Disentangling Successor Features for Coordination in Multi-agent Reinforcement Learning” at the *21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022)* [4]. A follow-on manuscript is in preparation for submission to the *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*.

An open question in MARL is how to ensure coordination among decentralized agents, particularly in unstructured tasks with sparse rewards and many agents — this problem is often described as one of “multi-agent credit assignment.” Our work aimed to address this problem by exploring whether *successor features (SFs) can be used to disentangle an individual agent’s impact on the global value function from that of all other agents, and if this disentanglement then enables better training of decentralized agents.* More specifically, we sought to understand if such disentanglement could improve decentralized training by enabling *compact* representation of a private utility that has high factoredness and *high learnability*. Our method uses three key ideas to address this question, all of which leverage the ability of SFs to separate environment dynamics from rewards.

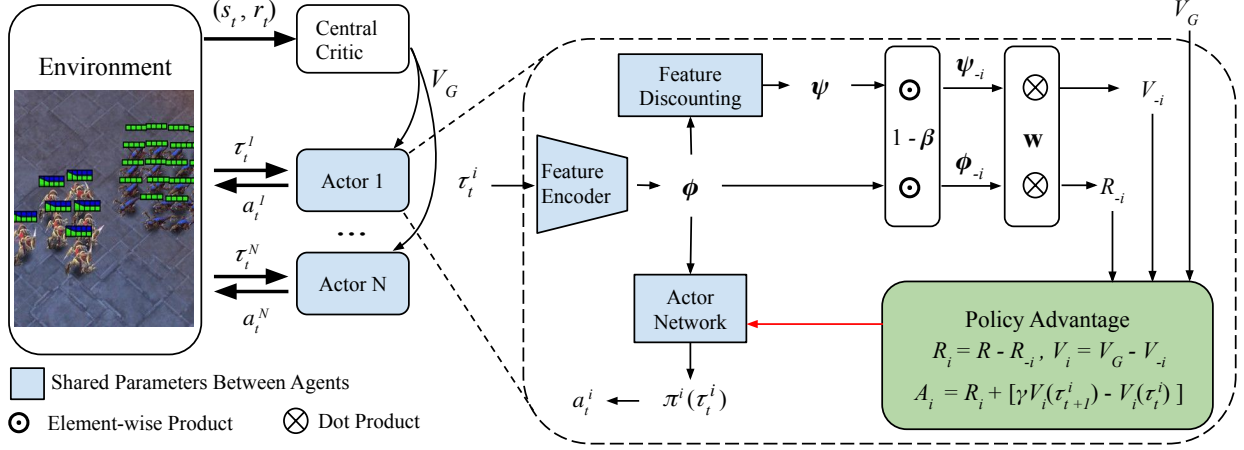


Figure 13: The architecture used to implement our DISSC framework. Our framework uses SFs to learn a disentanglement vector β that allows us to estimate individual agent value and reward functions, V_i and R_i , respectively. We use V_i and R_i to train decentralized agents, supported by a central critic used to estimate the global value function V_G . We optimize β for learnability, which we estimate using SFs, to improve training stability in complex environments with many agents.

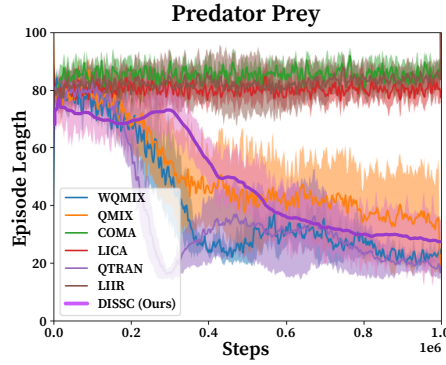


Figure 14: Training convergence plots in the predator-prey environment. Most methods solve the task with some variation in learning speed. Lower episode length indicates better performance, results are aggregated over 10 replicates.

First, we defined a private utility, based on a modified formulation of the estimated difference utility (EDU), that uses SFs to disentangle the impact of an agent from the global value function. We use our private utility, named SF-EDU, as an individual value function for training decentralized agents. Second, we introduced a process for learning the SF disentanglement vector used to calculate our SF-EDU. We optimize this disentanglement for learnability, which we estimate using SFs, to improve training stability and performance. Third, we introduced a centralized training, decentralized execution (CTDE) architecture with a shared feature encoding that efficiently implements these ideas.

We demonstrated our approach in three multi-agent environments: (1) a simple predator-prey environment, (2) the Starcraft Multi-Agent Challenge (SMAC), and (3) 5 vs. 5 and 7 vs. 7 CTF games. As seen in Figure 14, our method is able to solve the predator-prey task in a similar manner to other baselines, though slightly slower. We expect this difference is due to our method requiring learning of additional models not required by other methods, which do not provide performance

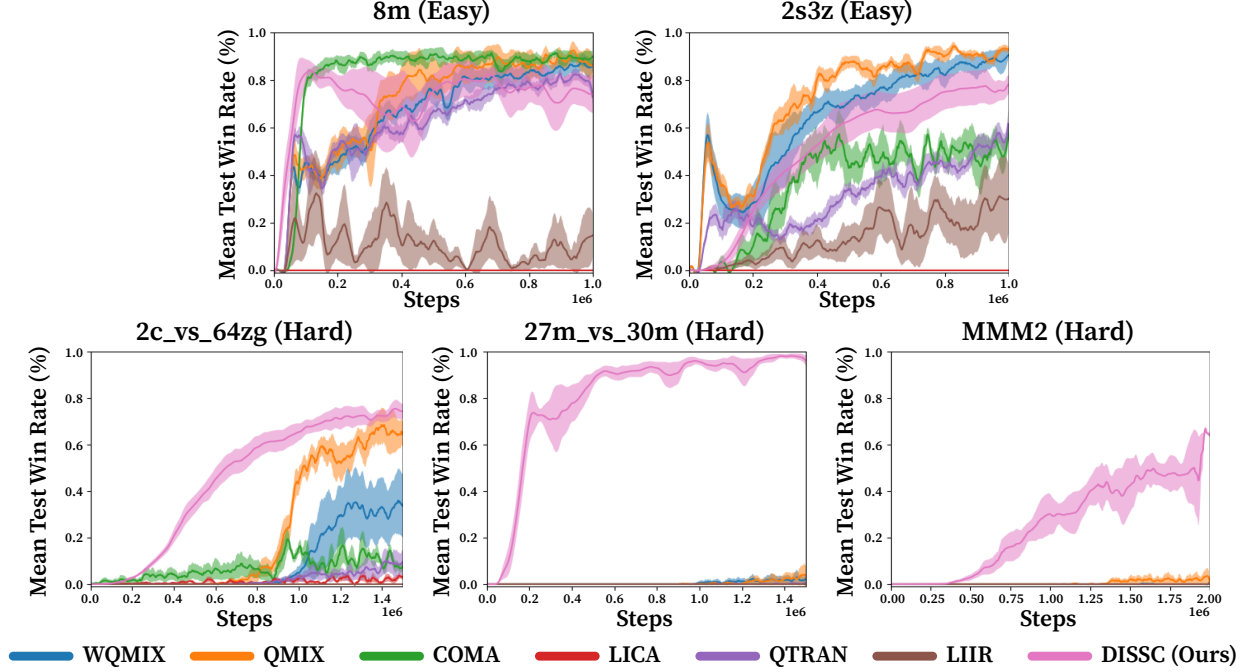


Figure 15: Training convergence plots in five SMAC maps, calculated over five replicates. A higher win-rate indicates better performance. In easier games, 8m and 2s3z, we match performance of the baselines. In harder games (especially 27m_vs_30m, in which we control 27 agents), our method has significantly faster learning times and higher performance.

benefits for a task as simple as predator-prey. However, for the more complex SMAC environment, Figure 15 shows that our method shows improved performance relative to baselines. For the 2c_vs_64zg map, only DISSC, QMIX, and WQMIX are able to obtain a non-trivial positive win-rate; all other methods essentially fail to solve the task. However, amongst these three, our method shows significantly faster learning, the highest converged performance, and the lowest variance in win rate. As we move to maps where more agents are controlled (i.e., 2c_vs_64zg, MMM2, 27m_vs_30m), we see that our method is the only one able to solve the task. We expect that these performance gains are due to the learning stability provided by our method through its direct optimization of learnability, which is most beneficial in complex environments with many agents.

To better understand our method, we also visualize the effect of our disentanglement vector β on individual agents, shown in Figure 16. We accomplish this by using a Grad-CAM-like approach to correlate the effect of observations to the latent space ψ . More specifically, we calculate a correlation, $\partial\psi/\partial o^i$, that captures the effect of changing the observation on the feature space. We then multiply this correlation by β to evaluate which elements of the observation space are being filtered by β .

We also compare our method to baselines in our CTF environment, to better understand how it performs in a more stochastic and sparse reward setting. We compare to WQMIX and QMIX (since they show the most promise among baselines in SMAC hard environments) in this experiment, along with a vanilla independent actor-critic (IAC) baseline. We experiment in 5-vs-5 and 7-vs-7 games, where teams are composed of two types of agents: slow, strong “convoy” agents and fast,

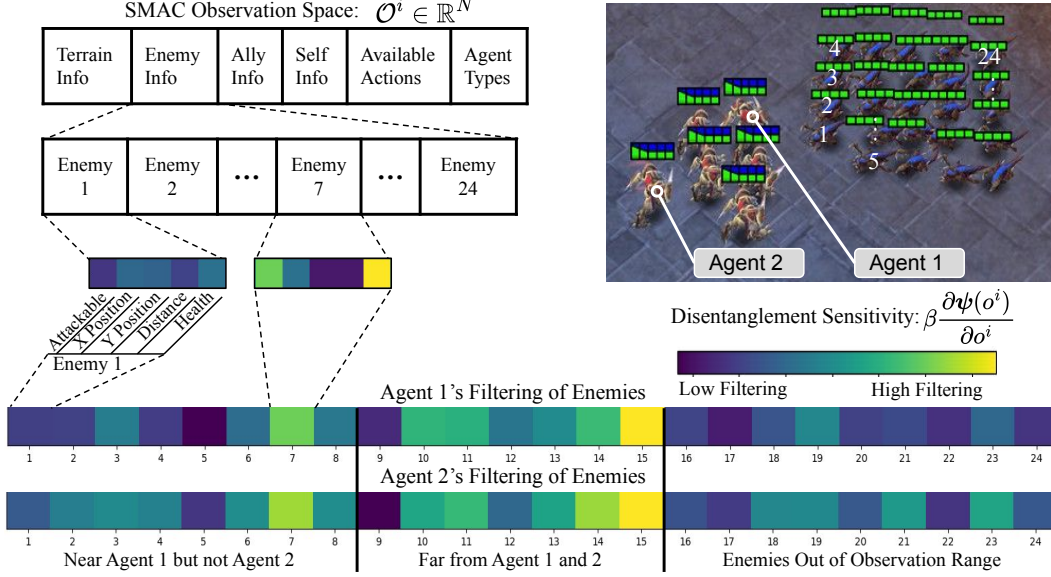


Figure 16: Effect of the disentanglement vector β on agent learning. Agent observations in SMAC are a vector with information about current position, enemies, allies, and available actions. We use Grad-CAM to correlate the sensitivity of the feature space, ψ , with respect to the observation, o^i . The gradient, $\partial\psi/\partial o^i$, can then be multiplied by the disentanglement vector, β , to understand which observation variables are the most heavily filtered during disentanglement. The breakout shows Agent 1 filters Enemy 1 less than Enemy 7, primarily by their “Attackable” and “Health” variables. This filtering occurs because Agent 1 is closest to Enemy 1 and can attack the enemy to generate reward by damaging them. The bottom breakout shows the summation of the filtering across all enemies for Agents 1 and 2 and supports the trend that enemies further away from agents are more heavily filtered.

weak “normal” agents. Figure 17 shows that our method, DISSC, significantly outperforms QMIX and WQMIX, with IAC falling somewhere in the middle. We expect that the poor performance of QMIX and WQMIX is due to convergence to a defensive policy where the agents remain on their own territory to limit their chances of being killed. This sub-optimal policy likely occurs due to limited exploration in this sparse reward environment, a known issue with many Q -learning methods. Figure 17d also contains results from an ablation study of DISSC, comparing the effects of our β disentanglement vector and our centralized critic on model performance. We see that removing β reduces training stability, supporting our assertion that our modified EDU improves training stability through its high learnability. We also see that removing the central critic, and instead allowing individual agents to evaluate the global value, results in the lowest converged global return within our model variants, suggesting that local SF critics struggle to estimate the global value. Local critics likely struggle because agents only have access to their own individual observations during training.

We also investigate how factoredness and learnability change during training to better understand our SF-EDU learning process, as shown in Figure 17e. We see that during training, DISSC is able to explicitly increase learnability of individual agents, though there appears to be a slight tradeoff between factoredness and learnability. This tradeoff is likely due to our approximation of the EDU, but is minimal relative to changes in learnability. These results suggest that learnability and factoredness may naturally conflict in some environments and therefore require simultaneous optimization of both in such settings.

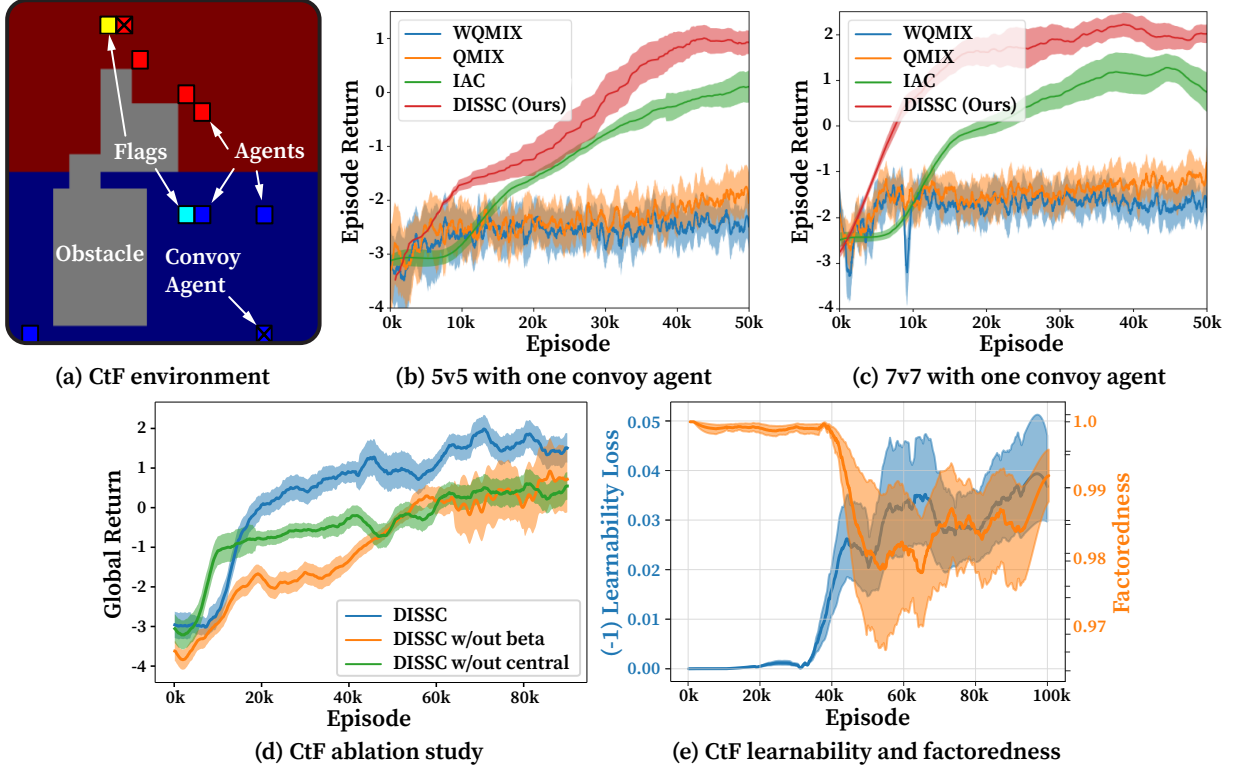


Figure 17: CTF experiment results. (a) The CTF environment. (b)-(c) Training convergence plots for our considered methods in two CTF settings, calculated over five replicates. Our method shows improved training time, converged performance, and stability over baselines. We also show (d) ablation results and (e) learnability and factoredness improvements during training.

Overall, our results suggest that disentanglement of SFs is a promising approach for improving performance in MARL relative to baseline methods, particularly in unstructured tasks with many agents, such as SMAC and CTF.

Accomplishment 8: A Framework for Minimizing Communications in MARL. Many MARL algorithms, including our own DISSC [4], assume a fully decentralized setting where agents have no communication. However, certain missions will allow some, but limited communications. We therefore created a new framework, named coordination machines, for creating a team of agents that minimize communications subject to a specified task completion success probability. This work was submitted as a manuscript titled “Coordination Machines for Minimizing Communication in Multi-Agent Reinforcement Learning” to the 2024 *Reinforcement Learning Conference (RLC) Workshop on Coordination and Cooperation in Multi-Agent Reinforcement Learning (Co-CoMARL)*, with a follow-on manuscript in preparation for submission to *IEEE Robotics and Automation Letters (RA-L)*.

More specifically, our goal was to design a method to train a team of agents and verify they can successfully complete a task with probability of at least p_c while minimizing inter-agent communication. We approached this problem by creating a hierarchical model, named a coordination machine, that decomposes a global, multi-agent task given by an LTL formula into an equivalent

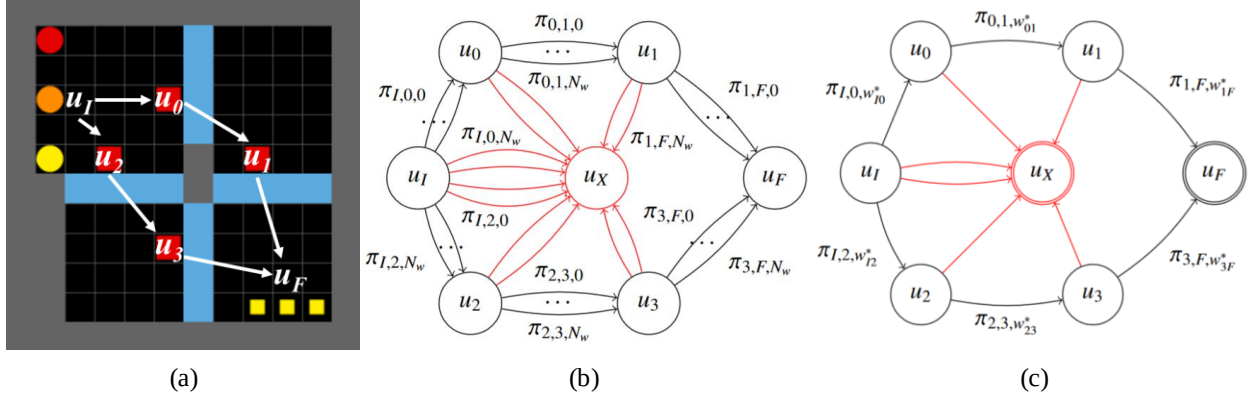


Figure 18: Our box lifting environment for testing our coordination machines framework. (a) agents must cooperate to reach the goal states in the bottom right room starting in the top left room, where blue doorways open after the red boxes are lifted. (b) the PFA representation of the task prior to selection of μ and ν . (c) the Markov chain induced on the PFA by selecting μ and ν .

automaton defined over subtasks given as LTL formulae. Our key idea is that each subtask may require different levels of inter-agent communication, and varying the level of communication allowed for each subtask can lower overall communication cost subject to a desired overall task success rate. Taking this hierarchical approach, we formulated an optimization problem to ensure a team of agents satisfies a probabilistic task-completion specification while minimizing inter-agent communication, and developed an algorithm to train a team of agents to complete each subtask using MARL. We formally defined a coordination machine as follows:

Definition 1. Coordination Machine A coordination machine is a tuple $\mathcal{C} = (\mathcal{T}, \mathcal{E}, \Pi, \mu, \nu)$ which satisfies $P(\diamond(u = u_F) \mid \mathcal{C}) \geq p_c$ while minimizing communication used by a low-level team of agents, where $\mathcal{T}$ is a PFA, $\mathcal{E}$ is a set of edges with properties $\{u'w, \phi_{u'}, \mathcal{M}_{uu'}, \pi_{uu'w}, p_{uu'w}\}$, Π is a set of low-level policies, μ is a high-level policy, and ν is a high-level policy-selection function.

We demonstrated our method in a simulated grid-world environment, shown in Figure 18, where the global task is for three agents to reach the goal states in the bottom right room. The agents begin in the top left room and progress through one of two possible paths, where each path requires that they cooperate to lift the red boxes which causes the blue doorways to open. This environment could model, for example, a team of robots jointly lifting large obstacles to get to a person trapped within a collapsed building. Figure 20 shows the success curves for trained agents solving each subtask (u_x) for various communication levels, specified by λ . We used the CASEC algorithm [8] to optimize agent policies, where λ defines the number of directed edges between agents for communication. Using these estimated success curves, we then solve our formulated optimization problem to identify the minimal communication cost possible for a range of p_c . Figure 20 shows these results. Note that the cost is guaranteed to monotonically increase as p_c increases since our method minimizes the level of communication needed to achieve a given p_c . We compare our formulation to a baseline formulation in which a single level of communication is used across all subtasks. This approximates a common formulation used in MARL works in which tasks are not decomposed into subtasks (i.e., they implicitly use a "monolithic", high-level model with a single transition that models the entire task) — our method differs in that we train

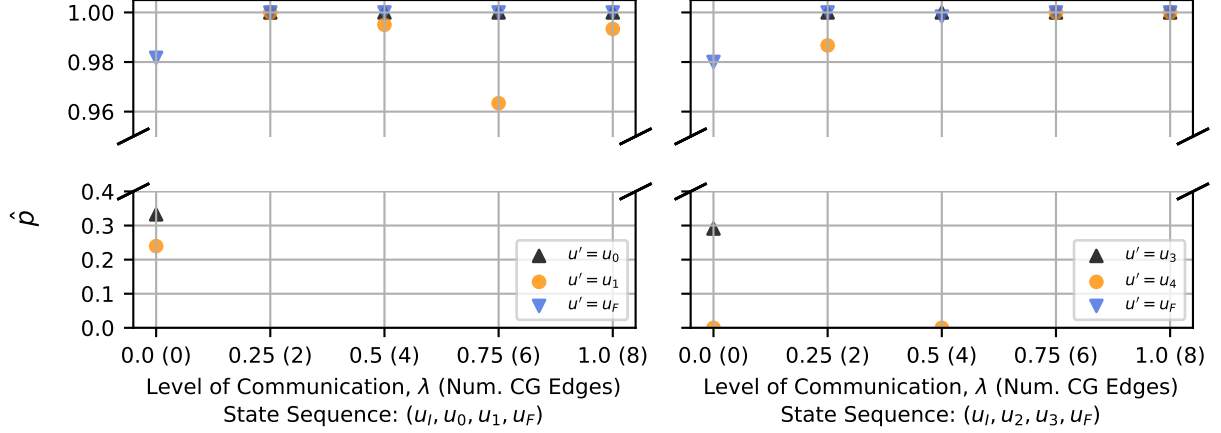


Figure 19: Estimated success probabilities for low-level policies. Each low-level policy was trained using CASEC [8] to complete a given subtask in the CM using a given level of communication, specified by λ . We see that while all subtasks are able to be completed with a high probability of success, changing the level of communication allowed can significantly affect that success probability.

different policies for each subtask instead of a single policy for the entire task, which allows us to vary the level of communication used within each subtask. We find that our method incurs a cost that is less than or equal to the baseline at all sampled values of p_c , with a 36% median reduction in cost across the sampled values of p_c and a maximum reduction in cost of 40% for $p_c = 0.9$. We observe that the baseline selects $\lambda_{u_I, u_0} = 0$ for $p_c \leq 0.98$, and $\lambda_{u_I, u_0} = 0.25$ for $p_c > 0.98$. This is because the baseline model has fewer degrees of freedom, and therefore cannot optimize communication based on the performance within each subtask.

Our results support the key thesis of this work, namely that decomposing a global task into subtasks enables us to ensure the MAS satisfies a given level of performance while minimizing inter-agent communication, while at the same time ensuring the global communication cost is less than or equal to a baseline monolithic model.

Accomplishment 9: Deploying SNN-based RL Policies on Neuromorphic Hardware for Low SWaP Applications. The majority of our work was demonstrated in simulated environments — we therefore also created methods for deploying RL policies on real robots. We focused on creating SNN-based RL policies that can be effectively deployed on neuromorphic hardware for potential deployment in low SWaP applications. This work is ongoing with a manuscript titled “SPAC-Net: Spiking Actor-Critic Network” in preparation for submission to *IEEE Robotics and Automation Letters (RA-L)*.

Due to the discontinuities introduced by the discrete spike events, SNNs cannot be trained by the typical back-propagation process used in traditional artificial neural networks (ANNs). We therefore explored the use of rate-based approximations (i.e., tuning curves). Given an arbitrary RL loss function, the main idea is to build a network of differentiable rate-based approximation units and solve for the synaptic weights by using an arbitrary version of gradient descent. Once the solution is obtained, the approximation units can be substituted with LIF neurons to use the

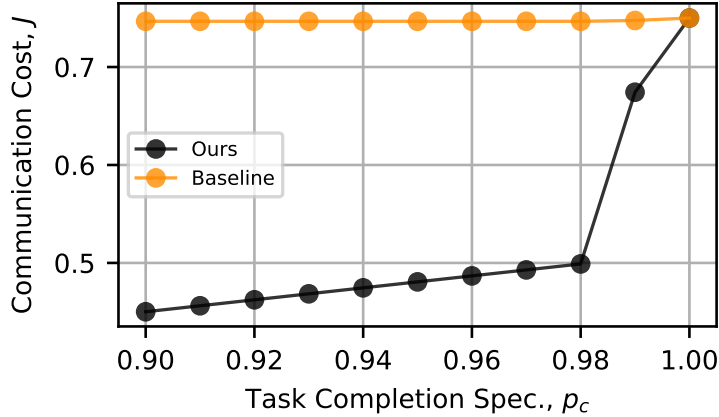


Figure 20: Global communication cost as a function of p_c . Our approach incurs a cost that is less than or equal to the baseline across all sampled values of p_c , with a 36% median reduction in cost across the sampled values of p_c .

resulting spiking network during inference. Our general approach is shown in Figure 21. We utilized NengoDL to implement our ANN to SNN conversion method. We used a standard PPO algorithm for training an ANN-based RL policy, which was then converted to an SNN form.

We tested our spiking PPO on the Loihi neuromorphic chip, a brain-inspired hardware designed to execute asynchronous spiking network with low energy consumption. While more advanced Loihi models are available on the Intel-INRC server, we used the Kapoho Bay model for its accessibility and potential deployment on mobile applications. We interfaced the Loihi chip with nengo-loihi and nxsdk to convert a trained policy network into spiking neurons and synapses. The conversion of an ANN to SNN RL policy involves re-compiling the non-linear activation functions as the node of the neuron, and defining synaptic connection scaled by weights. In our pipeline, the weight of the network is trained in a simulated SNN (sim-SNN) with a neuromorphic-compatible activation function (SoftLIF), and then converted to the on-chip network for evaluation.

We demonstrated our method in a 1 vs. 1 CTF game, and 2 vs. 2 CTF game, and the parking environment (shown in Figure 3). We tested both the simulated and the on-chip SNN agents in the same randomly initialized environment (using the same seeds) for 1000 trials. Table 3 shows our results, organized by four different outcomes: 1) Flag Capture: The SNN agent captures the enemy flag, leading to success; 2) Defeated: The SNN agent is intercepted by the enemy agent, resulting in failure; 3) Lost Flag: The enemy agent captures the friendly flag, resulting in failure for the SNN agent; 4) Time-up: The CTF episode ends upon reaching the maximum step limit. We see that our simulated SNN agent achieved a success rate of 78.1% in the 1 vs. 1 CTF game, with the on-chip agent achieving a slightly lower success rate of 72.6%. We note that, for both agents, the majority of the failure cases occurred when the enemy had advantageous initial conditions that either made it easier to defend its flag or faster to intercept our SNN agent. We see similar performance for the 2 vs. 2 CTF game and a lower success rate in the parking environment. Figure 22 shows a statistical comparison of our simulated SNN and Loihi-deployed SNN agents for the 1 vs. 1 CTF game. We see that the Loihi deployed agent has a much lower lower bound on performance, likely due to approximation errors of the simulated SNN; however, the medium and upper quantile performance

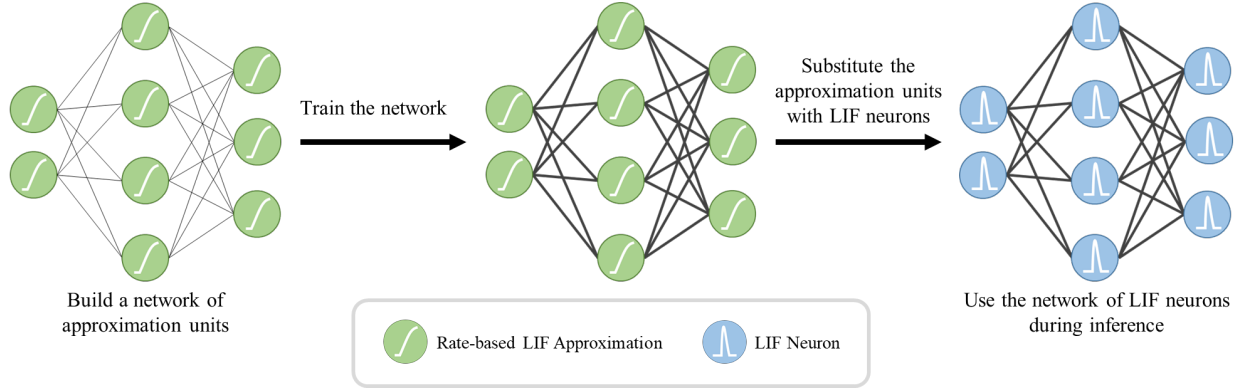


Figure 21: ANN-to-SNN conversion work-flow. We first employ rate-based approximations of the LIF neurons to build our network, since the discontinuous spike event outputs of the original LIF neurons prevents the training of the network through the back-propagation algorithm. After completing the training of this interim network, we substitute the LIF approximations with the original ones while keeping the network topology and the trained weights (bold black lines) the same.

Table 3: Results for our ANN-SNN converted RL agents.

Method	1v1 CTF				2v2 CTF				Parking	
	Flag Capture	Defeated	Lost Flag	Time-up	Flag Capture	Defeated	Lost Flag	Time-up	Success Rate	Crash Rate
sim SNN	78.1%	20.8%	0.4%	0.7%	82.3%	16.0%	0.6%	1.1%	43%	11%
Loihi	72.6%	26.4%	0.4%	0.6%	-	-	-	-		

results are similar for simulated and Loihi-deployed agents.

3.3 What opportunities for training and professional development did the project provide?

Eight graduate students, three undergraduate students, one research engineer, and one postdoctoral scholar were support in part by this project. These researchers received extensive one-on-one mentoring by the PIs on research methods, data analysis, experimental design, simulations, communicating technical work, and general career planning guidance through weekly meetings. They were also included in broader meetings with collaborators working on other autonomy and AI-related efforts, including UIUC faculty and research staff at the UIUC Applied Research Institute. These broader meetings exposed the researchers to other related ongoing work, giving them breadth in addition to the depth provided in weekly advising meetings. Graduate and undergraduate students were also able to present at and attend various conferences, which gave them opportunities to interact with their research communities and potential employers. More specifically, four graduate students and one undergraduate student presented at or attended at least one of the following conferences: the 2020 *IEEE International Conference on Robotics and Automation (ICRA)*, the 21st *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022)*, the *Adaptive and Learning Agents (ALA) Workshop at AAMAS 2022*, the 2023 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop on Advances in Multi-Agent Learning - Coordination, Perception, and Control*, and the 2023 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Two undergraduate students also participated in the

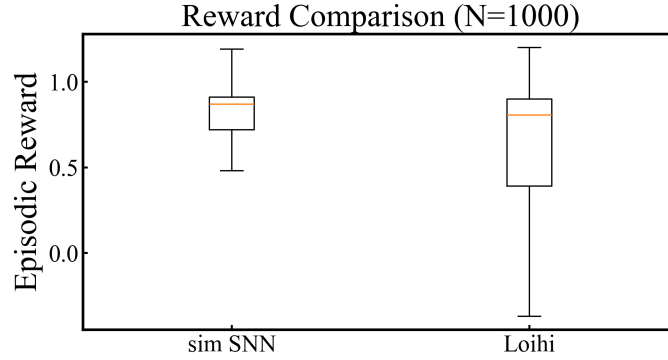


Figure 22: Episodic reward statistics for our trained simulated and on-chip SNN agents. The performance of the on-chip agent was slightly reduced due to approximation errors of the simulated SNN. The median reward values are illustrated by the orange lines, where the boxes depict the first and the third quartiles. We

project through the Undergraduate Research Opportunities Program (UROP) at UIUC, where they were able to present their results at the program’s end of summer presentation symposium. One PhD student successfully passed their preliminary exam, two PhD students successfully passed their qualifying exams, and one MS student successfully graduated with his MS degree. One PhD student obtained an internship at the U.S. Army Construction Engineering Research Laboratory (CERL), where he is working on applying methods developed from this project for CERL applications. One PhD student obtained a full-time position in industry. One MS student obtained a full-time position at the Naval Nuclear Laboratories.

3.4 How were the results disseminated to communities of interest?

The results from this project were disseminated to communities of interest through five manuscripts that were accepted or submitted to AI/robotics journals, conferences, or workshops [3, 4, 1, 2], with an additional eight manuscripts currently in preparation for submission. PI Tran also gave three invited talks on work from this project. One was titled “On Utilities for Cooperative Multi-agent RL” at the *IROS 2022 Decision making in Multi-agent Systems Workshop*. The second was titled “Learning for Decision Making in Cooperative Multi-agent Systems” at the *RSS 2023 Workshop on Inference and Decision Making for Autonomous Vehicles (IDMAV)*. The third was titled “Learning with Inductive Biases for Autonomous Decision-making” at the *GE Probabilistics Seminar Series*.

3.5 What honors or awards were received under this project?

Two PhD students working on this project were recipients of the *Aerospace Engineering Department Fellowship for outstanding academic and research achievement*. Another PhD student involved with the project received the *NASA Illinois Space Grant Consortium (ISGC) fellowship* and the *U.S. Department of Education’s Graduate Assistance in Areas of National Need (GAANN) fellowship*.

4 Technology Transfer

The work from this project led to several follow-on proposals to other DoD and civil agencies, including two of which were funded. The first funded follow-on effort was a project titled “RAAMM: Robust Adaptive Autonomy for Multi-agent Maneuvers” funded through the Army Research Lab (ARL) A2I2 program, and managed by Dr. Daniel Cassenti with Dr. John Rogers and Dr. Phil David as ARL collaborators. Our work on hierarchical RL and MARL contributed to that ARL project. The second funded follow-on effort is a project titled “Learning from Existing Systems for Emergent Coordination in Future Air Mobility” funded by NASA, and managed by Dr. Husni Idris. Our work on analyzing and discovering biases in multi-agent systems is contributing to that NASA project.

As mentioned in Section 3.3, one graduate student on this project also got an internship with the U.S. Army CERL, where he is working to apply the coordination machines framework developed for this project to their problems. Finally, the PIs had discussions with several other DoD agencies about transitioning the work done on this project, including Dr. Katarina Doctor from NRL, Dr. Michael Qin from ONR, and several members of Dr. Qin’s and Dr. Peter Squire’s performing teams (David Hattori-Messing from Leidos and Ethan Hada from the Naval Surface Warfare Center).

5 Participants

Participants for this project were Dr. Huy T. Tran (PI), Dr. Girish Chowdhary (co-PI), Dr. R.S. Sreenivas (co-PI), Dr. Heather Filippini (co-PI), Dr. Jagadeesh Yedetorre (co-PI), Dr. Aseem Borkar (Postdoctoral scholar), and Neale Van Stralen (Research Engineer).

6 Students

This project partially funded eight graduate STEM students, one of which has received a graduate STEM degree, and three undergraduate STEM students, one of which has received an undergraduate STEM degree. All students are on-track to receive their graduate or undergraduate STEM degrees.

7 Products

This project produced five manuscripts that were accepted or submitted to AI/robotics conferences or workshops [3, 4, 1, 2], with an additional eight manuscripts currently in preparation for submission.

References and Citations

- [1] M. Yuasa, H. T. Tran, and R. S. Sreenivas, “On Generating Explanations for Reinforcement Learning Policies: An Empirical Study,” Sept. 2023. arXiv:2309.16960 [cs].
- [2] R. Nigam, N. Parikh, M. Yuasa, and H. T. Tran, “Coordination in Ad Hoc Teams with Generalized Policy Improvement,” in *IROS 2023 Workshop on Advances in Multi-Agent Learning - Coordination, Perception, and Control*, (Detroit, MI), 2023.
- [3] N. V. Stralen, S. Kim, H. T. Tran, and G. Chowdhary, “Feature Specialization and Clustering Improves Hierarchical Sub-task Learning,” in *Proc. of the Adaptive and Learning Agents Workshop (ALA 2022)*, 2022.
- [4] S. H. Kim, N. V. Stralen, G. Chowdhary, and H. T. Tran, “Coordination in Multi-agent Reinforcement Learning with Successor Features,” in *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022)*, (London, UK), 2022.
- [5] X. Li, Z. Serlin, G. Yang, and C. Belta, “A formal methods approach to interpretable reinforcement learning for robotic planning,” *Science Robotics*, vol. 4, no. 37, pp. 1–16, 2019.
- [6] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. Van Hasselt, and D. Silver, “Successor features for transfer in reinforcement learning,” in *Advances in Neural Information Processing Systems*, pp. 4056–4066, 2017. arXiv: 1606.05312 ISSN: 10495258.
- [7] A. J. Plantinga, R. Walsh, and M. Wibbenmeyer, “Priorities and Effectiveness in Wildfire Management: Evidence from Fire Spread in the Western United States,” *Journal of the Association of Environmental and Resource Economists*, vol. 9, pp. 603–639, July 2022.
- [8] T. Wang, L. Zeng, W. Dong, Q. Yang, Y. Yu, and C. Zhang, “Context-Aware Sparse Deep Coordination Graphs,” in *ICLR 2022*, arXiv, June 2022. arXiv:2106.02886 [cs].

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.			
1. REPORT DATE (DD-MM-YYYY) 07-05-2024	2. REPORT TYPE Final Performance Report	3. DATES COVERED (From - To) April 2020 - December 2023	
4. TITLE AND SUBTITLE Explainable AI for Mission Planning and Execution with Interpretable Courses of Action		5a. CONTRACT NUMBER N00014-20-1-2249	
		5b. GRANT NUMBER GRANT12954047	
		5c. PROGRAM ELEMENT NUMBER 	
6. AUTHOR(S) Tran, Huy, T. Chowdhary, Girish Sreenivas, R. S. Filippini, Heather		5d. PROJECT NUMBER 	
		5e. TASK NUMBER 	
		5f. WORK UNIT NUMBER 	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) UNIVERSITY OF ILLINOIS 506 S WRIGHT ST URBANA, 61801-3620, USA		8. PERFORMING ORGANIZATION REPORT NUMBER 	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research 875 N. Randolph Street, Suite 1425 Arlington, VA 22203-1995		10. SPONSOR/MONITOR'S ACRONYM(S) ONR	
		11. SPONSORING/MONITORING AGENCY REPORT NUMBER 	
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution is Unlimited.			
13. SUPPLEMENTARY NOTES 			
14. ABSTRACT The overall goal of this project was to support Navy planning and execution by developing methods for improved explainability, interpretation, and performance in artificial intelligence (AI) systems. This project was organized into three primary research thrusts. Thrust one developed methods to identify coherent COAs from AI systems, including a heuristic temporal logic search algorithm and a neurosymbolic approach for learning general temporal logic formulae from agent trajectories. Thrust two developed methods to discover biases from decision sequences and incorporate biases into trained decision policies, including a framework for an RL agent to adapt to biased teammates, methods for analyzing the impact of agent biases on team performance, and a survey of inverse reinforcement learning (RL) methods for discovering agent biases. Thrust three developed methods to improve the performance of explainable RL agents, including a method for training hierarchical RL agents that can handle adversaries, an RL method for training large teams of coordinated agents in adversarial environments, a framework for minimizing communications in multi-agent RL, and a method for deploying spiking neural network-based RL policies on neuromorphic hardware for low SWaP applications. Our results were disseminated to communities of interest through AI/robotics journals, conferences, workshops, and invited seminars.			
15. SUBJECT TERMS Artificial intelligence, Autonomy, Machine learning, Reinforcement learning, Explainable AI			

INSTRUCTIONS FOR COMPLETING SF 298

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER <i>(Include area code)</i>

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g., 30-06-1998; xx-08-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. 1F665702D1257.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. AFOSR-82-1234.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORS AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.