Toward Policy Explanations for Multi-Agent Reinforcement Learning

Kayla Boggess[1] , **Sarit Kraus**[2] and **Lu Feng**[1]

[1]University of Virginia
[2]Bar-Ilan University

{kjb5we, lu.feng}@virginia.edu, sarit@cs.biu.ac.il

Abstract

Advances in multi-agent reinforcement learning (MARL) enable sequential decision making for a range of exciting multi-agent applications such as cooperative AI and autonomous driving. Explaining agent decisions is crucial for improving system transparency, increasing user satisfaction, and facilitating human-agent collaboration. However, existing works on explainable reinforcement learning mostly focus on the single-agent setting and are not suitable for addressing challenges posed by multi-agent environments. We present novel methods to generate two types of policy explanations for MARL: (i) policy summarization about the agent cooperation and task sequence, and (ii) language explanations to answer queries about agent behavior. Experimental results on three MARL domains demonstrate the scalability of our methods. A user study shows that the generated explanations significantly improve user performance and increase subjective ratings on metrics such as user satisfaction.

1 Introduction

Recent years have witnessed a growing body of research in multi-agent reinforcement learning (MARL), enabling sequential decision making for a range of exciting multi-agent applications such as cooperative AI [Dafoe *et al.*, 2020] and autonomous driving [Kiran *et al.*, 2021]. Generating explanations about agent decisions is crucial for improving system transparency, increasing user satisfaction, and facilitating human-agent collaboration [Kraus *et al.*, 2020; Chakraborti *et al.*, 2020]. However, existing works on explainable reinforcement learning (RL) mostly focus on the single-agent setting [Wells and Bednarz, 2021; Heuillet *et al.*, 2021; Puiutta and Veith, 2020].

Generating explanations for MARL agents that interact with each other in a common environment poses significant challenges. The combinatorial nature of MARL (i.e., the joint state/action space grows exponentially with the number of agents) leads to scalability issues. Explanations should provide adequate information about agent behavior, including the interaction (e.g., cooperation) among multiple agents, for user understanding. Furthermore, explanations should avoid redundant information that may overwhelm or confuse users, thus decreasing user satisfaction and trust.

To tackle these challenges, we develop novel methods to generate two types of policy explanations for MARL: (i) policy summarization, and (ii) query-based language explanations. Our methods rely on first building an abstract representation of MARL policy as a multi-agent Markov decision process (MMDP), which can be obtained by abstracting samples observed during the MARL policy evaluation. The proposed method generates a summarization about the most probable sequence of agent behavior under a given MARL policy, by finding the most probable path through the MMDP abstraction and extracting information about agent cooperation and task sequence. The generated policy summarizations can help users to have a global view of agent decisions and support human-agent collaboration (e.g., users may adjust their workflow based on agents' task sequence).

Additionally, we developed methods for generating language explanations to answer three types of queries about agent behavior, including "When do [agents] do [actions]?", "Why don't [agents] do [actions] in [states]?", "What do [agents] do in [conditions]?" Such explanations can enable users to understand specific agent decisions, debug faulty agent behavior, and refine user mental models. Our work is inspired by the method proposed in [Hayes and Shah, 2017], which computes a minimal Boolean logic expression covering states satisfying the query criteria, and converts the Boolean expression to explanations via language templates. However, we find that a naive adaptation of this method for MARL generates explanations with redundant information and has limited scalability. Further, the generated explanations do not necessarily capture agent cooperation, which is imperative for explaining MARL. We proposed improved methods that address these limitations by leveraging MARL domain knowledge (e.g., agent cooperation requirements) to filter relevant agent states and actions.

We applied a prototype implementation of the proposed methods to three benchmark MARL domains: (i) multi-robot search and rescue, (ii) multi-robot warehouse, and (iii) level-based foraging [Papoudakis *et al.*, 2021]. Experimental results demonstrate that the proposed methods can generate policy summarizations and query-based explanations for large MARL environments with up to 19 agents.

Finally, we conducted a user study to evaluate the quality

of generated explanations. We measured user performance on correctly answering questions based on explanations, to test user understanding of agent behavior. We also collected user subjective ratings on *explanation goodness metrics* [Hoffman *et al.*, 2018]. The results show that the generated explanations significantly improve user performance and increase subjective ratings on various metrics including user satisfaction.

2 Related Work

Explaining agent decision making has recently emerged as a focus area within the explainable AI paradigm. [Chakraborti *et al.*, 2020] provides a survey about this emerging landscape of explainable decision making. [Kraus *et al.*, 2020] proposes Explainable Decisions in Multi-Agent Environments (xMASE) as a new research direction, emphasizing many challenges of generating multi-agent explanations, such as accounting for agent interactions and user satisfaction.

Explainable RL has been attracting increasing interest, as shown in several recent surveys [Wells and Bednarz, 2021; Heuillet *et al.*, 2021; Puiutta and Veith, 2020]. In particular, [Wells and Bednarz, 2021] points out the lack of user studies as a major limitation across existing works. Moreover, current approaches mostly focus on the single-agent setting, while generating explanations for MARL has received scant attention so far. [Kazhdan *et al.*, 2020] extracts MARL model as abstract argumentations, which do not explicitly consider agent cooperation on the same tasks as in our work.

The proposed methods are inspired by several prior works. [Topin and Veloso, 2019] develops Abstracted Policy Graphs (i.e., Markov chains of abstract states) for explaining single-agent RL. Our work differs in that we adopt MMDP to represent multi-agent policy abstractions. [Amir and Amir, 2018] summarizes agent behavior by extracting trajectories from agent simulations and visualized them as videos. Inspired by this idea, we created GIF animations as the baseline for evaluating policy summarizations in the user study. But we extracted the sequence of agent actions differently, focusing on the agent cooperation. [Hayes and Shah, 2017] generates RL policy descriptions to answer queries. We adapted this method to multi-agent environments and proposed significant improvements, which will be described in Section 4.

3 Policy Abstraction and Summarization

We describe how to build an abstract representation of MARL policy in Section 3.1, and present the method for generating policy summarization in Section 3.2.

3.1 Policy Abstraction

In the context of MARL, a group of N agents interact with each other in a common environment and make decisions influenced by the joint states of all agents. Agent decisions can be captured by a joint policy $\pi : \mathcal{X} \rightarrow \Delta(\mathcal{A})$, which is a function mapping the set of joint states $\mathcal{X} = \{(x^1, \ldots, x^N)\}$ to a probabilistic distribution over the set of joint actions $\mathcal{A} = \{(a^1, \ldots, a^N)\}$, where x^i (*resp.* a^i) denotes the state (*resp.* action) of agent i. Once a policy is trained, agents can act upon it in any given state. But there is a lack of a global view of the entire policy. Further, the size of the policy grows
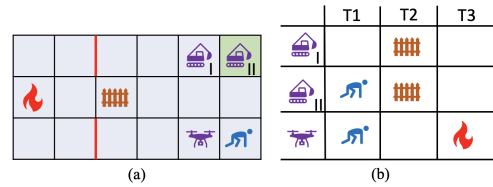

Figure 1: Example MARL domain of multi-robot search and rescue.

exponentially with the number of agents and state variables. To address these issues, we propose to build an abstract representation of the policy as the basis for generating explanations about agent behavior.

We use the multi-agent Markov decision process (MMDP) framework to represent MARL policy abstraction. Formally, an MMDP is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T})$, where $\mathcal{S} = \{(s^1, \ldots, s^N)\}$ is the joint (abstract) state space, $\mathcal{A}$ is the joint action space, and $\mathcal{T}$ is the transition function. Let $\mathcal{F}$ be a set of Boolean predicates indicating features of the MARL domain. We denote by $f(x^i) = 1$ if an agent state x^i satisfies a feature predicate $f \in \mathcal{F}$. An abstract state s^i is then given by the satisfaction of all feature predicates $f \in \mathcal{F}$, where each bit of the binary encoding of $s^i \in \mathbb{N}$ corresponds to the satisfaction of a predicate $f(x^i)$. Thus, the choice of features affects the abstraction level and should include adequate information for explanations. In this work, we assume that users specify a set of feature predicates for a given MARL domain.

Once an MARL policy is trained, we build an MMDP during the policy evaluation stage. For each sample $(\mathbf{x}, \mathbf{a}, \mathbf{x}')$, determine an MMDP transition $\mathbf{s} \xrightarrow{\mathbf{a}} \mathbf{s}'$ by finding the abstract state $\mathbf{s}$ (*resp.* $\mathbf{s}'$) corresponding to $\mathbf{x}$ (*resp.* $\mathbf{x}'$). When policy evaluation terminates (e.g., converging to the expected reward), compute the transition probability $\mathcal{T}(\mathbf{s}, \mathbf{a}, \mathbf{s}')$ via frequency counting.

Properties. The resulting MMDP is a *sound* abstraction of the MARL policy because, by construction, every MMDP transition with non-zero probability corresponds to at least one sampled policy decision. The state space size $|\mathcal{S}|$ is bounded by $\mathcal{O}(2^{|\mathcal{F}|^N})$, depending on the number of agents N and feature predicates $|\mathcal{F}|$. In practice, a trained MARL policy may only induce a small set of reachable states.

Example 1 *Figure 1(a) shows an example MARL domain where three robotic agents cooperate to complete search and rescue tasks. Rescuing the victim requires the cooperation of an unmanned aerial vehicle (UAV) and an unmanned ground vehicle (UGV). Any agent can fight the fire, which is blocked by the wall and obstacle. Removing the obstacle requires the cooperation of two UGVs. Given a trained MARL policy, we build an MMDP abstraction with 6 feature predicates indicating whether each task is detected or completed (e.g.,* victim_detect, *victim_complete). An agent can only detect a task in a neighboring grid (e.g., UAV detects the victim in Figure 1(a)). The resulting MMDP has 63 (reachable) states and 577 transitions.*

3.2 Policy Summarization

A policy abstraction containing hundreds of states and transitions is too complex for humans to understand. An alterna-

---

Algorithm 1 Generating Policy Summarization

Input: policy abstraction $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T})$, task completion predicates $\mathcal{F}_c \subseteq \mathcal{F}$
Output: policy summarization $\mathcal{Z}$
1: $\mathcal{Z} \leftarrow \{\}$
2: Compute the most probable path ρ through $\mathcal{M}$
3: **for** $0 \leq t \leq |\rho|$ **do**
4:      $y \leftarrow$ new array
5:      **for** $1 \leq i \leq N$ **do**
6:          $y[i] \leftarrow \{\}$
7:          **for** $f \in \mathcal{F}_c$ **do**
8:             **if** agent state s_t^i in the path ρ satisfies f **then**
9:                 insert f to $y[i]$
10:      insert non-empty array y to $\mathcal{Z}$
11: **return** $\mathcal{Z}$

---

tive way of communicating agent behavior is to show execution traces; however, a lengthy trace may be burdensome for users to review. To overcome these limitations, we develop a method to generate policy summarization, illustrating the agent cooperation and task sequence for the most probable sequence of agent behavior under a given MARL policy.

Algorithm 1 shows the proposed method, which takes the input of a policy abstraction $\mathcal{M}$ and a set of predicates $\mathcal{F}_c$ representing the completion of tasks (subgoals) in a given MARL domain. The first step is to compute the most probable path $\rho = \mathbf{s}_0 \xrightarrow{\mathbf{a}_0} \mathbf{s}_1 \xrightarrow{\mathbf{a}_1} \cdots$ from the initial state to a goal state in the MMDP $\mathcal{M}$, which represents the most probable sequence of agent decisions under the policy. This problem can be solved by converting the MMDP to a directed weighted graph with edge weight $e(\mathbf{s}, \mathbf{a}, \mathbf{s}') = -\log \mathcal{T}(\mathbf{s}, \mathbf{a}, \mathbf{s}')$ for each transition, and then applying the Dijkstra's algorithm [Dijkstra *et al.*, 1959] to find the shortest path.

Next, the algorithm loops through every joint state $\mathbf{s}_t$ in the path ρ to extract the agent cooperation and task sequence. At each step t, the algorithm checks if an agent state s_t^i satisfies any task completion predicate $f \in \mathcal{F}_c$ and inserts completed task f into the array element $y[i]$ (line 4-9). An agent only satisfies a task completion predicate at step t when it finishes the task and receives a reward. We assume that if a task is completed via the cooperation of multiple agents, they must satisfy the task predicate f at the same step t and each receive a portion of the reward. Thus, the agent cooperation is represented as multiple elements of the array y sharing the same task. Only non-empty arrays containing completed tasks are inserted into the summarization $\mathcal{Z}$. When the algorithm terminates, the generated summarization is visualized as a chart, with each column corresponding to a non-empty y-array and each row representing an agent's task sequence.

Properties. The generated policy summarization $\mathcal{Z}$ is *sound*, because it is derived from the most probable path of a *sound* policy abstraction (see Section 3.1). The complexity of computing the most probable path is bounded by $\mathcal{O}(|\mathcal{S}|^2)$, following the complexity of the Dijkstra's algorithm and depending on the MMDP state space size. The rest of Algorithm 1 is bounded by $\mathcal{O}(|\rho| \cdot N \cdot |\mathcal{F}_c|)$, depending on the path length and the number of agents and tasks.

Example 2 *We apply Algorithm 1 using the policy abstraction and task predicates from Example 1. There are 8 states in the most probable path from the initial state (i.e., all agents starting in the green grid) to a goal state (i.e., all tasks have*

been completed). Figure 1(b) visualizes the generated summarization, with column names (i.e., T1, T2, T3) indicating the sequence of task completions: UGV_2 and UAV cooperate to rescue the victim; next, UGV_1 and UGV_2 cooperate to remove the obstacle; and lastly, UAV fights the fire.

4 Query-Based Explanations

While policy summarization provides a global view of the agent behavior under a MARL policy, users may also query about specific agent decisions. In this section, we develop methods for generating language explanations to answer the following three types of queries:

- *"When do [agents] do [actions]?"* for identifying conditions for action(s) of a single or multiple agent(s) .

- *"Why don't [agents] do [actions] in [states]?"* for understanding differences in expected and observed behaviors of a single or multiple agent(s).

- *"What do [agents] do in [predicates]?"* for revealing agent behavior under specific conditions described by the given predicates.

Our work is inspired by a method developed in [Hayes and Shah, 2017] to generate query-based explanations for single-agent RL. In the following, we propose new methods to tackle limitations posed by adapting this baseline method to multi-agent environments.

4.1 Explanations for When Query

Algorithm 2 presents both the baseline and proposed methods for answering "When do agents G_q do actions A_q?", where G_q and A_q are sets of agents and actions, respectively. The text in blue highlights changes about relevancy filters (RF) for the proposed method (called WithRF) compared to the baseline (called NoRF).

WithRF starts the algorithm (line 1-5) by identifying relevant agents G, features F, and action sets A based on domain knowledge (e.g., agent cooperation requirements). For example, consider a query "When does UAV rescue the victim?". The domain knowledge is that rescuing the victim requires the cooperation of a UAV and a UGV (Example 1). Thus, the relevant agent set G is {UVA, UGV_1, UGV_2}. The relevant feature set F is {victim_detect, victim_complete}, while predicates about the fire and obstacle are irrelevant. The relevant action sets A is an array with each element representing one possible set of agent actions required for cooperation: [{UAV_rescue, UGV_1_rescue}, {UAV_rescue, UGV_2_rescue}], which can be generated based on the aforementioned domain knowledge about agent cooperation requirements.

Both NoRF and WithRF loop through all the joint states $\mathbf{s} \in \mathcal{S}$ of the policy abstraction MMDP and check all the *enabled* (i.e., with non-zero transition probability) joint actions $\mathbf{a}$ in state $\mathbf{s}$. In line 9 of Algorithm 2, NoRF checks if $\mathbf{a}$ is *compatible* with A_q; that is, every agent action $a \in A_q$ is contained in the joint action $\mathbf{a} = (a^1, \ldots, a^N)$. By contrast, WithRF checks if $\mathbf{a}$ is compatible with at least one set of relevant actions contained in the array A. Following the previous example, NoRF checks if $\mathbf{a}$ contains UAV_rescue, while WithRF checks if $\mathbf{a}$ contains {UAV_rescue, UGV_1_rescue} *or*

Algorithm 2 Generating Query-Based Explanations

Input: policy abstraction $(\mathcal{S}, \mathcal{A}, \mathcal{T})$, query "when do agents G_q do actions A_q?"
Output: explanations $\mathcal{E}$
1: $G \leftarrow \{\}; F \leftarrow \{\}; A \leftarrow [\{\}]$
2: **for all** agent action $a^i \in A_q$ **do**
3:      insert all relevant agents of a^i to G
4:      insert all relevant features of a^i to F
5:      insert all relevant action sets of a^i to A
6: $V \leftarrow \{\}; \bar{V} \leftarrow \{\}$
7: **for all** joint state $\mathbf{s} \in \mathcal{S}$ **do**
8:      **for all** joint action $\mathbf{a}$ enabled in $\mathbf{s}$ **do**
9:          **if** $\mathbf{a}$ is compatible with A_q [replace A_q with A] **then**
10:              insert $\mathbf{s}$ to V
11:          **else**
12:              insert $\mathbf{s}$ to $\bar{V}$
13: $B_1 \leftarrow \text{States2Boolean}(V); B_0 \leftarrow \text{States2Boolean}(\bar{V})$
14: $\phi \leftarrow \text{Quine-McCluskey(ones=}B_1\text{, zeros=}B_0\text{)}$
15: translate ϕ to explanations $\mathcal{E}$ via language templates
16: **return** $\mathcal{E}$
17: **function** STATE2BOOLEAN(W)
18:      $B \leftarrow \{\}$
19:      **for all** $\mathbf{s} = (s^1, \dots, s^N) \in W$ **do**
20:          **for** $1 \leq i \leq N$ [replace with $i \in G$] **do**
21:              $C \leftarrow$ feature predicate valuations of state s^i
22:              **for** $f \in \mathcal{F}$ [replace with $f \in F$] **do**
23:                  insert $C(f)$ to B
24:      **return** B

$\{\text{UAV\_rescue}, \text{UGV}_2\text{\_rescue}\}$. Since each element of A is a super-set of A_q, the WithRF check is more restrictive.

If a state $\mathbf{s}$ has at least one enabled action $\mathbf{a}$ passing the aforementioned checks, $\mathbf{s}$ is inserted to the target states set V; and to the non-target states set $\bar{V}$ otherwise. The intuition is that the generated explanations should describe target states satisfying the query criteria and exclude conditions of non-target states. WithRF poses further restrictions that target states need to satisfy criteria captured by relevant actions A, such as agent cooperation requirements. Thus, explanations generated by WithRF can provide information about agent cooperation, which may be missed by NoRF explanations.

Next, the algorithm converts the states set V (*resp.* $\bar{V}$) to a list of Boolean formulas B_1 (*resp.* B_0) via the function described in line 17-24. Given a joint state $\mathbf{s} = (s^1, \dots, s^N)$, NoRF finds valuations of every feature predicates $f \in \mathcal{F}$ for all agent state s^i and insert them to the list B. By contrast, WithRF only inserts to B the valuations of relevant features $f \in F$ in relevant agent states s^i for all $i \in G$. Following the previous example, WithRF only considers Boolean formulas related to relevant feature predicates $\{\text{victim\_detect}, \text{victim\_complete}\}$, filtering out features related to the fire and obstacle.

Lastly, the algorithm supplies Boolean formulas B_1 and B_0 to the Quine-McCluskey algorithm [Quine, 1952] and obtains a minimized Boolean formula, which can be translated into language explanations following [Hayes and Shah, 2017]. The runtime of Quine-McCluskey grows exponentially with the number of variables. Thus, WithRF is more efficient than NoRF, due to the decreased number of Boolean variables. Moreover, filtering out irrelevant agents and features helps WithRF to prevent redundant information in the generated explanations.

Properties. Following the Quine-McCluskey, the complexity of NoRF is bounded by $\mathcal{O}\big(3^{N \cdot |\mathcal{F}|} / \ln(N \cdot |\mathcal{F}|)\big)$. The complexity of WithRF is reduced to $\mathcal{O}\big(3^{|G| \cdot |F|} / \ln(|G| \cdot |F|)\big)$.

Example 3 *Table 1 (first row) shows the explanations generated by NoRF and WithRF for a when query. The NoRF explanation contains redundant information about the fire and obstacle that are irrelevant to the query. The WithRF explanation completely captures the required agent cooperation for the query, which is missed by the NoRF explanation.*

4.2 Explanations for Why Not Query

The query "Why don't agents G_q do actions A_q in the joint State $\mathbf{s}_q$?" can be answered by modifying Algorithm 2 as follows. In line 10, adding $\mathbf{s}$ to $\bar{V}$ instead of V. Remove line 11-12 and add a new line for inserting the query state $\mathbf{s}_q$ to V. The modified algorithm (see Appendix A in [Boggess *et al.*, 2022]) generates an explanation describing the differences between the observed behavior in the target query state $\mathbf{s}_q$ and the expected behavior of states with actions compatible with the query actions A_q (NoRF) or relevant action sets A (WithRF). The complexity of the modified algorithm follows Algorithm 2.

Example 4 *Table 1 (second row) shows the explanations generated by NoRF and WithRF for a why not query about the behavior of two agents UGV_1 and UGV_2 in the state shown in Figure 1(a). The WithRF explanation captures the required agent cooperation for removing the obstacle, while the NoRF explanation fails to provide such information.*

4.3 Explanations for What Query

To answer the query "What do agents G_q do when satisfying predicates F_q?", we first identify all the satisfying joint states $\mathbf{s} = (s^1, \dots, s^N)$; that is, for all $i \in G_q$, agent state s^i satisfies predicates F_q. The baseline NoRF method is to generate a list of all possible enabled actions for agents G_q in these states. The proposed WithRF method improves the baseline by filtering agent actions that are relevant to predicates F_q and finding the most likely relevant agent actions from the list via frequency counting. Since the proposed methods[1] do not need to call the Quine-McCluskey, the complexity of both NoRF and WithRF are only bounded by $\mathcal{O}(|G_q| \cdot |\mathcal{S}| \cdot |\mathcal{A}|)$, depending on the number of query agents, joint state space, and joint action space of the policy abstraction.

Example 5 *Table 1 (third row) shows the explanations generated by NoRF and WithRF for a what query. The WithRF explanation is more concise than the NoRF explanation and only contains relevant action for the query predicate.*

5 Computational Experiments

We implemented and applied the proposed methods to three MARL domains. The first domain is multi-robot search and rescue (SR) similar to Example 1. The second and third domains are benchmarks taken from [Papoudakis *et al.*, 2021]. Multi-robot warehouse (RWARE) considers multiple robotic agents cooperatively delivering requested items. Level-based foraging (LBF) considers a mixed cooperative-competitive game where agents must navigate a grid world to collect randomly scattered food. Our implementation used the Shared

---

[1] See Appendix A in [Boggess *et al.*, 2022] for the pseudos code.

Query	Explanations generated by NoRF (baseline)	Explanations generated by WithRF (proposed)
When does UAV rescue the victim?	UAV rescues the victim when UAV detects the victim *and* UGV_1 does not detect the fire, *or* UAV detects the victim *and* UGV_1 does not detect the obstacle.	UAV rescues the victim when UAV detects the victim *and* UGV_1 detects the victim, *or* UAV detects the victim *and* UGV_2 detects the victim.
Why don't UGV_1 and UGV_2 remove the obstacle in this state?	UGV_1 and UGV_2 don't remove the obstacle in this state because UGV_1 does not detect the obstacle.	UGV_1 and UGV_2 don't remove the obstacle in this state because UGV_1 does not detect the obstacle *and* UGV_2 does not detect the obstacle.
What does UAV do when it detects the victim?	UAV can rescue the victim, move, or wait when it detects the victim.	UAV is most likely to rescue the victim when it detects the victim.

Table 1: Examples of query-based explanations

Case Study		MMDP		Summarization		When Query				Why Not Query				What Query			
				Path	Chart	NoRF		WithRF		NoRF		WithRF		NoRF		WithRF	
Domain	N	$|\mathcal{S}|$	$|\mathcal{T}|$	$|\rho|$	$|\mathcal{Z}|$	$|\mathcal{E}|$	Time (ms)	$|\mathcal{E}|$	Time (ms)	$|\mathcal{E}|$	Time (ms)	$|\mathcal{E}|$	Time (ms)	$|\mathcal{E}|$	Time (ms)	$|\mathcal{E}|$	Time (ms)
SR	3	63	577	8	3x2	4	72.4	4	1.2	1	86.8	2	0.8	3	1.7	1	1.3
	4	732	5,048	23	4x4	-	timeout	6	31.4	-	timeout	3	14.9	3	15.7	1	14.5
	5	839	4,985	24	5x4	-	timeout	10	73.6	-	timeout	4	23.4	6	24.6	1	19.8
RWARE	2	8	62	5	2x2	4	2.6	4	2.7	4	2.0	2	0.4	3	0.1	1	0.1
	4	16	387	5	4x1	12	3,321.0	5	82.7	6	23.0	3	1.3	3	0.1	1	0.1
	19	114	1,500	15	19x2	-	timeout	7	3,890.6	-	timeout	4	56.5	3	21.7	1	20.1
LBF	2	5	13	4	2x2	2	0.7	2	0.7	3	0.4	2	0.3	2	0.1	1	0.1
	4	15	355	5	4x1	10	3,598.5	2	1.4	8	3,695.0	2	1.6	3	0.6	1	0.6
	9	482	5,841	13	9x2	-	timeout	2	200.2	-	timeout	2	101.3	2	24.9	1	19.8

Table 2: Experimental results on three MARL domains (timeout set as one hour).

Experience Actor-Critic [Christianos *et al.*, 2020] for MARL policy training and evaluation. All models were trained and evaluated to 10,000 steps, or until converging to the expected reward, whichever occurred first. The experiments were run on a laptop with a 1.4 GHz Quad-Core Intel i5 processor and 8 GB RAM.

Table 2 shows experimental results. For each domain, we report the number of agents N and the number of states $|\mathcal{S}|$ and transitions $|\mathcal{T}|$ of generated policy abstraction MMDP. It is unsurprising that the MMDP size grows exponentially with the number of agents. We report the most probable path length $|\rho|$ and the chart size $|\mathcal{Z}|$ of generated policy summarizations, which are more compact and easier to interpret than complex MMDP abstractions. All summarizations were generated within 1 second (thus not shown in the table). Additionally, we compare NoRF and WithRF methods in terms of the number of clauses in the generated query-based explanations and runtime. The results show that WithRF is more succinct in general and has better scalability than NoRF. In particular, NoRF failed to generate explanations for "when" and "why not" queries within an hour for large cases with more than 4 agents, while WithRF generated explanations for all cases within seconds. Both methods generate explanations for "what" queries efficiently, thanks to the lower complexity than other queries (see Section 4).

In summary, experimental results demonstrate that the proposed methods can generate policy summarizations and query-based explanations for large MARL domains (e.g., RWARE with 19 agents, which is the largest number of possible agents in the provided environments).

6 User Study

We conducted a user study [2] via Qualtrics to evaluate the quality of generated explanations. We describe the study design in Section 6.1 and analyze results in Section 6.2.

6.1 Study Design

Participants. We recruited 116 eligible participants (i.e., fluent English speakers over the age of 18) through university mailing lists. 62.1% of participants self-identified as male, 37.1% as female, and 0.8% preferred not to say. The age distribution is 76(18-24), 31(25-34), 7(35-49), 2(50-64). Participants were instructed to answer multiple-choice questions about agent behavior for multi-robot search and rescue tasks. They were incentivized with bonus payments to answer questions correctly based on the provided explanations. To ensure data quality, attention checks were injected during the study.

Independent variables. We employed a within-subject study design with the explanation generation methods as independent variables. Participants were asked to complete two trials for evaluating policy summarizations. They were presented with charts generated by Algorithm 1 in one trial, and GIF animations illustrating the most probable sequence of agent behavior (i.e., visualization of the most probable path in the policy abstraction) in the other trial. For each trial, there were two questions about agent behavior in various environments (i.e., 3×6 and 6×6 grid world). Questions used in the two trials are different but had similar difficulty. All participants were presented with the same set of four randomly generated questions for summarization trials. To counterbalance the ordering confound effect, they were randomly assigned to answer the first two questions based on either charts or GIF, and the other two questions based on the remaining method. Additionally, participants were asked to complete two trials for evaluating query-based explanations generated by NoRF and WithRF methods, with 6 questions (2 environments × 3 query types) in each trial. Participants answered the same set of 12 randomly generated questions for query-based trials, and were randomly assigned to different groups similarly to summarization trials.

Dependent measures. We measured *user performance* by counting the number of correctly answered questions in each trial. In addition, at the end of each trial, participants were asked to rate in a 5-point Likert scale (1 - strongly disagree,
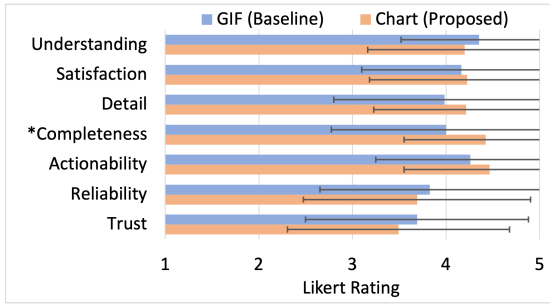
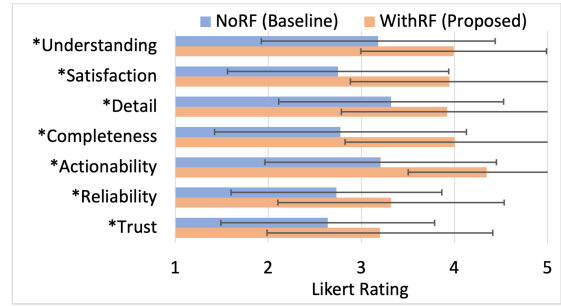Figure 2: Mean and SD of participant ratings about policy summarizations ("*" indicates statistically significant difference).

Figure 3: Mean and SD of participant ratings about query-based explanations ("*" indicates statistically significant difference).

5 - strongly agree) about *explanation goodness* metrics (i.e., understanding, satisfaction, detail, completeness, actionability, reliability, trust) [Hoffman *et al.*, 2018].

Hypotheses. We make the following hypotheses in this study.

- **H1:** Chart-based summarizations lead to better user performance than GIF-based.

- **H2:** Chart-based summarizations yield higher user ratings on explanation goodness metrics than GIF-based.

- **H3:** Query-based explanations generated by WithRF lead to better user performance than those by NoRF.

- **H4:** Query-based explanations generated by WithRF yield higher user ratings on explanation goodness metrics than those by NoRF.

6.2 Results Analysis

We used a paired t-test to evaluate hypotheses H1 and H3, and used the Wilcoxon Signed-rank test to evaluate hypotheses H2 and H4. We set the significant level as $\alpha = 0.05$.

Evaluating policy summarizations. Participants answered more questions correctly with chart-based summarizations (M=1.8 out of 2, SD=0.6) than GIF-based (M=0.9 out of 2, SD=0.4), with statistical significance ($t(462)$=-15.8, $p \leq 0.01$, d=1.5). *Thus, the data supports H1.*

Figure 2 shows average participant ratings about summarizations. Chart-based summarizations yield higher ratings on the perceived completeness than GIF-based with statistical significance (W=371.5, Z=-2.4, $p \leq 0.02$, r=-0.2). But no significant difference was found regarding the other metrics. *Thus, the data partially supports H2.*

Evaluating query-based explanations. Participants answered more questions correctly with explanations generated by WithRF (M=5.2 out of 6, SD=1.7) than NoRF (M=2.3 out of 6, SD=1.0), with statistical significance ($t(1390)$=-21.1, $p \leq 0.01$, d=2.0). *Thus, the data supports H3.*

Figure 3 shows that participants gave higher average ratings to WithRF explanations than NoRF explanations. The Wilcoxon test found significant differences on all metrics: understanding (W=319.5, Z=-4.9, $p \leq 0.01$, r=-0.3), satisfaction (W=266.0, Z=-7.0, $p \leq 0.01$, r=-0.5), detail (W=484.0, Z=-3.7, $p \leq 0.01$, r=-0.2), completeness (W=494.5, Z=-6.4, $p \leq 0.01$, r=-0.4), actionability (W=167.0, Z=-6.9, $p \leq 0.01$, r=-0.5), reliability (W=382.5, Z=-3.6, $p \leq 0.01$, r=-0.2), and

trust (W=217.0, Z=-3.4, $p \leq 0.01$, r=-0.2). *Thus, the data supports H4.*

Discussion. In summary, the data supports all hypotheses, while H2 is only partially supported because the statistical test found no significant differences between chart-based and GIF-based summarizations on most metrics. However, Figure 2 shows that participants rated chart-based summarizations close to 4 (agree) on all metrics, and above GIF-based ratings on all metrics except understanding, reliability, and trust. This may be because users showed a strong preference toward the moving nature of GIF animations and the visualized effects of agents completing tasks. But watching a GIF can be more time-consuming and less informative than a quick glance at the chart. This is supported by the results that participants were able to answer more questions correctly with chart-based summarizations, and they rated this method significantly higher on completeness (i.e., providing needed information). Meanwhile, query-based explanations generated by the proposed WithRF method led to significantly better user performance and higher user ratings on all metrics, because users prefer succinct WithRF explanations with adequate information about agent behavior and cooperation. By contrast, NoRF explanations do not necessarily provide essential information about agent cooperation for correctly answering questions, and may contain redundant information that decreases user satisfaction.

7 Conclusion

In this work, we developed methods to generate policy summarizations and query-based explanations for MARL. Experimental results on three MARL domains demonstrate the scalability of our methods. Evaluation via a user study shows that our generated MARL policy explanations can improve user understanding about agent behavior and enable them to answer more questions correctly, while maintaining very positive ratings on explanation goodness metrics.

The proposed methods are independent from the inner working of MARL methods, only relying on policy abstractions that can be built via observing samples during the policy evaluation. Although we only used one MARL method in our experiments, the proposed methods can be used to generate policy explanations for different MARL methods. As part of the future work, we plan to apply the proposed methods to a wide range of MARL methods and domains.

Acknowledgments

References

[Amir and Amir, 2018] Dan Amir and Ofra Amir. Highlights: Summarizing agent behavior to people. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi Agent Systems*, pages 1168–1176, 2018.

[Boggess *et al.*, 2022] Kayla Boggess, Sarit Kraus, and Lu Feng. Toward policy explanations for multi-agent reinforcement learning. *arXiv preprint arXiv:2204.12568*, 2022.

[Chakraborti *et al.*, 2020] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. The emerging landscape of explainable automated planning & decision making. In *IJCAI*, pages 4803–4811, 2020.

[Christianos *et al.*, 2020] Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. In *Thirty-fourth Conference on Neural Information Processing Systems*, pages 10707–10717. Curran Associates Inc, 2020.

[Dafoe *et al.*, 2020] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.

[Dijkstra *et al.*, 1959] Edsger W Dijkstra, others, et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[Hayes and Shah, 2017] Bradley Hayes and Julie A Shah. Improving robot controller transparency through autonomous policy explanation. In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*, pages 303–312. IEEE, 2017.

[Heuillet *et al.*, 2021] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Explainability in deep reinforcement learning. *Knowledge-Based Systems*, 214:106685, 2021.

[Hoffman *et al.*, 2018] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*, 2018.

[Kazhdan *et al.*, 2020] Dmitry Kazhdan, Zohreh Shams, and Pietro Liò. Marleme: A multi-agent reinforcement learning model extraction library. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[Kiran *et al.*, 2021] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[Kraus *et al.*, 2020] Sarit Kraus, Amos Azaria, Jelena Fiosina, Maike Greve, Noam Hazon, Lutz Kolbe, Tim-Benjamin Lembcke, Jorg P Muller, Soren Schleibaum, and Mark Vollrath. Ai for explaining decisions in multi-agent environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13534–13538, 2020.

[Papoudakis *et al.*, 2021] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[Puiutta and Veith, 2020] Erika Puiutta and Eric MSP Veith. Explainable reinforcement learning: A survey. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 77–95. Springer, 2020.

[Quine, 1952] Willard V Quine. The problem of simplifying truth functions. *The American mathematical monthly*, 59(8):521–531, 1952.

[Topin and Veloso, 2019] Nicholay Topin and Manuela Veloso. Generation of policy-level explanations for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2514–2521, 2019.

[Wells and Bednarz, 2021] Lindsay Wells and Tomasz Bednarz. Explainable ai and reinforcement learning—a systematic review of current approaches and trends. *Frontiers in artificial intelligence*, 4:48, 2021.