# Transparency Communication for Reinforcement Learning in Human-Robot Interaction

**David V. Pynadath**[1], **Ning Wang**[1], **Michael J. Barnes**[2],
[1] Institute for Creative Technologies, University of Southern California
[2] US Army Research Laboratory
pynadath@usc.edu, nwang@ict.usc.edu, michael.j.barnes.civ@mail.mil

## Abstract

Autonomous systems need to learn from their experience to improve their decisions. Machine-learning methods, particularly reinforcement learning, have successfully applied quantitative probabilities and utilities to a variety of real-world domains, including allowing robots to improve from their mistakes. The complexity of the domains that robots operate in and the inherent complexity of their decisions make understanding the inner workings of robotics systems increasingly challenging for their human teammates. While transparency communications have shown to alleviate such problems, the additional capability of self-improving, enabled by reinforcement learning, is likely to complicate the robot's effort to reason transparently with human teammates. In this paper, we discuss the design of both model-based and model-free reinforcement learning for the robots in a human-robot simulation testbed and the design of transparency communications that unpacks the components of the robot's decision-making and learning process.

## 1 Introduction

Autonomous systems operating in complex domains must reason under uncertainty, prioritize conflicting goals, and adapt to unpredictable environments. While autonomous systems have increased their capabilities over the decades, their potential is not often realized when teamed up with humans [Parasuraman and Riley, 1997]. Due to the complexity of the domains that autonomous systems now operate in and the inherent complexity of the decisions they make, understanding the inner workings of such systems becomes increasingly challenging for the humans. Such "black box" phenomena often result in disuse or over-reliance of the autonomous systems [Parasuraman and Riley, 1997]. In human-human teams, communication targeting the gaps in understanding can help improve shared situation awareness, foster trust relationships, and enhance team performance. Inspired by the research in human-human teams, researchers have designed both handcrafted and automatically generated explanations to achieve similar outcomes in human-automation teams [Dzindolet *et al.*, 2003; Wang *et al.*, 2016].

However, the autonomous systems used in transparency communication research often lack the ability to learn from their experience or from data to improve their decisions. The very definition of autonomy, in the field of artificial intelligence, means that a system should function beyond the initial knowledge encoded by the designers and must be able to adapt in the face of the unforeseen and the unknown [Russell and Norvig, 2016]. Machine-learning methods have been widely applied to the design of automation, including virtual agents and robots. Reinforcement Learning (RL), in particular, is a powerful machine-learning method that has successfully applied quantitative probabilities and utilities to a variety of real-world domains [Kaelbling *et al.*, 1996]. RL is based on the idea that an agent can autonomously discover an optimal behavior through trial-and-error interactions with its environment. The agent explores the space of possible strategies and receives feedback on the outcomes of the choices made. From this information, a "good"—or ideally optimal—policy (i.e., strategy or controller) can be deduced [Sutton and Barto, 1998; Kober *et al.*, 2013]. RL's algorithms for computing long-term expected values can provide autonomous agents with optimal sequential decision policies.

While RL is based on the simple idea of action-reward mappings, elements of RL can be complex due to their quantitative values and the iterative process by which they are computed. The rich representation and complex reasoning from RL, which provide useful performance guarantees for software agents, also present a significant obstacle to human understanding of, for example, how the value functions are constructed, how the algorithms update the value function, and how such updates impact the action/policy chosen by the agent. Without such understanding, a human operator is likely to fall into the same pitfall of misuse or disuse of the agents [Parasuraman and Riley, 1997]. Therefore, for autonomous learning agents to play an effective role in human-machine teams, they must make their RL-generated decisions understood by their human operators and teammates.

Automatically generated explanations have provided such understanding in other areas of artificial intelligence (AI) [Swartout *et al.*, 1991], suggesting the potential for similar understanding of RL-based AI. Model-based RL first learns a quantitative model in the form of Partially Observable Markov Decision Processes (POMDPs), which contain probabilistic action and sensor models, utility-based goal priorities, etc. [Kaelbling *et al.*, 1998] that could facilitate explanations. For real-world domains, the size and complexity of quantitative models like POMDPs are more likely to overwhelm human operators, rather than inform them. In our previous work, we have developed algorithms to automatically generate explanations for decisions made by POMDP-based agents. These algorithms were among the first research efforts to apply the methodology of explainable AI (XAI) to make quantitative agent-based decision models explainable and transparent to human users [Wang *et al.*, 2016].

In this paper, we discuss the design of RL for the robots in our human-robot reconnaissance task. We then discuss the design of explanations that can potentially make RL components transparent to human teammates.

## 2 Related Work

As AI systems become increasingly complex and bestowed with the capability to learn and evolve, their decision-making mechanisms become increasingly opaque to their human users. With the maturation of many AI-powered technologies, they are transitioning from operating from behind-the-scenes, to directly interfacing with humans. The "black box" issue of many AI systems has hindered the growth of the capability of human-AI systems teams. This problem has garnered much attention, and much research effort has been exerted to create an interface to make the decision process of AI algorithms more transparent. For example, in [Hendricks *et al.*, 2016], researchers used a combination of convolutional neural networks (CNN) and recurrent neural networks (RNN) to recognize objects in images and generate image captions based on recognized objects. Other efforts have focused not on unpacking the "black box" itself, but on generating explanations of decisions (e.g., classifications) from the "black box" based on the input (e.g., instances from dataset) and the output [Ribeiro *et al.*, 2016].

Our current work follows a long history of automated XAI mechanisms, especially within the context of expert systems [Swartout and Moore, 1993]. While most of this work operated on rule- and logic-based systems, there has been more recent work on generating explanations based on Markov Decision Problems (MDPs) [Elizalde *et al.*, 2008] and Partially Observable MDPs (POMDPs) [Wang *et al.*, 2016]. The existing evidence is encouraging as to the potential success of applying a general-purpose explanation on top of an agent's decision-making process. In addition to explaining POMDP-based decisions in simulated robots, there has been recent work on generating explanations and justifications for decisions by simulated and physical robots using decision trees [Sheh, 2017], answering human queries of the robot's policies [Hayes and Shah, 2017], and making the robot's planning process easier for humans to understand and predict [Zhang *et al.*, 2017].

There is a rapidly growing body of work on applying reinforcement learning to enable robots to improve from their mistakes, e.g., [Matarić, 1997; Smart and Kaelbling, 2002]. While such learning is likely to complicate the robot's effort to reason transparently with human teammates, it does provide an opportunity to repair trust that has been damaged by robot errors. Our investigation into the interaction between explanations and trust repair is inspired by work on methods for the latter within organizations [Lewicki, 2006; Schweitzer *et al.*, 2006]. Prior research has similarly examined the effectiveness of these trust-repair strategies within HRI [Robinette *et al.*, 2015]. This work found that the timing and combination of trust-repair actions was critical to effectively maintaining trust.

As RL has moved into more and more domains, there have also been more and more investigations into XAI with RL systems. For example, in [Iyer *et al.*, 2018], researchers produced visualizations of the state and behavior from agents using deep reinforcement learning networks. Within a specifically human-robot domain, researchers proposed Instruction-based Behavior Explanation (IBE) to allow human users to interactively provide input into the RL process while receiving explanations of the estimated outcome of such input. [Fukuchi *et al.*, 2017]. In our previous work, we have experimented with robot transparency communications that acknowledges errors and promises to improve [Wang *et al.*, 2018]. Such communication did not influence human trust and team performance, possibly due to a lack of explanation of *how* the robot plans to improve its decisions. In the work presented here, we focus on applying model-based and model-free reinforcement learning to update (and hopefully improve) the robot's decision-making and on designing explanations that communicate such an updating process to the human in an effort to build transparency and repair trust.

## 3 Human Robot Interaction Testbed

We conduct our investigation of explainable AI within an online HRI testbed that we have used to gather human behavior data when interacting with a simulated robot [Wang *et al.*, 2015].

### 3.1 POMDP Model of Testbed

The robot in this testbed bases its decisions on a POMDP [Kaelbling *et al.*, 1998] model, which is a tuple, $\langle S, A, P, \Omega, O, R \rangle$, that we describe here in terms of the HRI testbed scenario [Wang *et al.*, 2015]. In it, a human teammate works with a robot in reconnaissance missions to gather intelligence in a foreign town. Each mission involves the human teammate searching buildings in the town. The robot serves as a scout, scans the buildings for potential danger, and relays its observations to the teammate. Prior to entering a building, the human teammate can choose between entering with or without equipping protective gear. If there is danger present inside the building, the human teammate will be injured if not wearing the protective gear, causing the team to incur a high time penalty. However, it also takes time to put on and take off protective gear (although much less time than the injury penalty). So the human teammate is incentivized to consider the robot's observations before deciding how to enter the building. The simulated robot has an NBC (nuclear, biological and chemical) weapon sensor, a camera that can detect hostile gunmen, and a microphone that can listen to discussions in foreign language.

The state, $S$, consists of objective facts about the world, some of which may be hidden from the agent itself. We use a *factored representation* [Boutilier *et al.*, 2000] that decomposes these facts into individual feature-value pairs, such as the robot's current location, as well as the presence of dangerous people or chemicals in the buildings to be searched. The state may also include feature-value pairs that represent the health level of its human teammate, any current commands, and the accumulated time cost so far.

The available actions, $A$, correspond to the possible decisions the agent can make. Given the proposed mission, the agent's first decision is where to move to next. Upon completing a search of a building, an agent can make a decision as to whether to declare a location as safe or unsafe for its human teammate. For example, if a robot believes that armed gunmen are at its current location, then it will want its teammate to take adequate preparations (e.g., put on body armor) before entering. Because there is a time cost to such preparations, the robot may instead decide to declare the location safe, so that its teammate can more quickly complete their own reconnaissance tasks.

The transition probability function, $P$, represents the effects of the agent's actions on the subsequent state. In the current testbed, the robot's movement actions always succeed. Recommendation actions, on the other hand, can affect the health and happiness of its human teammate, although only stochastically, as a person may not follow the recommendation.

The "partial observability" of a POMDP is specified through a set of possible observations, $\Omega$, that are probabilistically dependent (through the observation function, $O$) on the true state of the world. Different observations may have different levels of noise. For example, an agent may be able to use GPS to get very accurate readings of its own location. However, it may not be able to detect the presence of hostile gunmen or dangerous chemicals with perfect reliability or omniscience. Instead, the agent will receive local readings about the presence (or absence) of threats in the immediate vicinity. For example, if dangerous chemicals are present, then the robot's chemical sensor may detect them with a high probability. There is also a lower, but nonzero, probability that the sensor will not detect them. In addition to such a false negative, we can also model a potential false positive reading, where there is a low, but nonzero, probability that it will detect chemicals even if there are none present. By controlling the observations that the agents receive, we can manipulate their ability level in our testbed.

Partial observability gives the robot only a subjective view of the world, where it forms beliefs about what it thinks is the state of the world, computed via standard POMDP state estimation algorithms. For example, the robot's beliefs may include its subjective view on the presence of threats, in the form of a likelihood (e.g., a 33% chance that there are toxic chemicals in the farm supply store). Again, the robot would derive these beliefs from prior beliefs about the presence of such threats, updated by its more recent local sensor readings. Due to the uncertainty in its prior knowledge and sensor readings (not to mention its learning), the robot's beliefs are likely to diverge from the true state of the world. By decreasing the accuracy of the robot's observation function, $O$, we can decrease the accuracy of its beliefs, whether receiving correct or incorrect observations. In other words, we can also manipulate the robot's ability by allowing it to over- or under-estimate its sensors' accuracy.

The human-machine team's mission objectives are captured by the reward function, $R$, which maps the state of the world into a real-valued evaluation of benefit for the agents. In our example domain, the robot receives the highest reward when the surveillance is complete. It will also receive higher reward values when its teammate is alive and unharmed. This reward component punishes the agents if they fail to warn their teammates of dangerous buildings. Finally, the agent will receive a slight negative reward for the amount of time that passes. This motivates the agents to complete the mission as quickly as possible.

By constructing such a POMDP model of the mission, the agent can autonomously generate its behavior by determining the optimal action based on its current beliefs, $b$, about the state of the world [Kaelbling *et al.*, 1998]. The agent uses a (bounded) lookahead procedure that seeks to maximize expected reward by simulating the dynamics of the world from its current belief state across its possible action choices. It will combine these outcome likelihoods with its reward function and choose the option that has the highest expected reward.

### 3.2 Explanation Generation for POMDPs

The elements $\langle S, A, P, \Omega, O, R\rangle$ of a POMDP model correspond to concepts that people are likely to be familiar with. By exposing different components of an agent's model, we can make different aspects of its decision-making transparent to human teammates. In prior work, we created static templates to translate the contents of a POMDP model into human-readable sentences [Wang *et al.*, 2016]. We create such templates around natural-language descriptions of each state feature and action. We then instantiate the templates at runtime with prespecified functions of the agent's current beliefs (e.g., probability of a state feature having a certain value). The following list illustrates the templates we created for each POMDP component, using specific runtime instantiations to show the final natural-language text provided to a human participant:

***S*:** The agent can communicate its current beliefs about the state of the world, e.g., "I believe that here are no threats in the market square." The agent could also use a standard POMDP probabilistic belief state to communicate its uncertainty in that belief, e.g., "I am 67% confident that the market square is safe."

***A*:** An agent can make a decision about what route to take through its search area, e.g., "I am proceeding through the back alley to the market square."

***P*:** An agent can also reveal the relative likelihood of possible outcomes based on its transition probability model, e.g., "There is a 33% probability that you will be injured if you follow this route without taking the proper precautions."

- ***Ω*:** Communicating its observation can reveal information about an agent's sensing abilities, e.g., "My NBC sensors have detected traces of dangerous chemicals."
- ***O*:** Beyond the specific observation it received, an agent can also reveal information about its observation model, e.g., "My image processing will fail to detect armed gunmen 30% of the time."
- ***R*:** By communicating the expected reward outcome of its chosen action, an agent can reveal its alignment (or lack thereof) with the mission objective, contained in its reward function, e.g., "I think it will be dangerous for you to enter the informant's house without putting on protective gear. The protective gear will slow you down a little." The template here relies on factored rewards, allowing the agent to compute separate expected rewards, $E[R]$, over the goals of keeping its teammate unharmed and achieving the mission as quickly as possible.

### 3.3 Results from POMDP-Based Explanation

We have used this testbed to gather human behavior data when interacting with the robot under combinations of these POMDP-based explanation algorithms. We designed explanations that selected different aspects of the robot's decision-making process at different level of details. Evaluations of such explanations have indicated that when the robot provides an explanation to justify its decisions (e.g., by providing a numerical confidence level of its decisions or a summary of findings from its sensors), the human-robot team performed better on simulated missions, compared to when no explanations were given [Wang *et al.*, 2016].

More relevant to a learning robot, we augmented explanations with a trust-repair strategy inspired by prior work in organizational trust: an acknowledgement of a mistake, paired with a promise to improve before the next mission [Schweitzer *et al.*, 2006]. This acknowledgment made the human teammate trust the robot more when the robot did not provide any explanation of its previous (and mistaken) decision. However, overall, such communication did not make any impact on the team performance, across a variety of explanation content [Wang *et al.*, 2018]. This lack of impact was most likely due to the fact that participants interacted with each specific robot for only one mission, so they would never be able to observe any improvement (so we never implemented any either). The natural next step is to enrich our agent to perform this self-improvement during the mission itself. This result highlighted the importance of unpacking the machine learning process of the robot and making it transparent to a robot's human teammate.

## 4 Reinforcement Learning in the HRI Testbed

### 4.1 Model-Based RL

Given the original POMDP model, it is a relatively easy step for the agent to update that model using reinforcement learning. As in most RL domains, we assume that $S$, $A$, and $\Omega$ are known a priori. That leaves $P$, $O$, and $R$ as the functions to be learned. In this domain, $R$ is a pre-defined mission objective, so no learning occurs there.

The robot's movement is deterministic, so the only part of the transition probability, $P$, to be learned is the probability that the teammate will follow the robot's recommendation. In the decision cycle after giving its recommendation, the robot observes whether its human teammate followed it or not. It can then use this signal to modify the probability table within $P$ corresponding to its recommendation action and the subsequent effect on the teammate's life. For example, if the teammate ignored the robot's recommendation to wear protective gear and was injured as a result, then the robot could decrease $P(\neg \textit{human injured}, \textit{recommend protective gear}, \neg \textit{human injured})$ and increase $P(\neg \textit{human injured}, \textit{recommend protective gear}, \textit{human injured})$. We do not experiment with such learning, as such a change would make the robot less likely to recommend protective gear in the future, whereas it should instead attempt an alternative explanation to convince the teammate to follow its recommendation.

We instead focus our model-based RL on the observation function, $O$. In many of our prior experiments, we have given the robot a broken camera that fails to detect gunmen in a building, even when present. Allowing such a robot to update $O$ based on experience would help it overcome such a lack of reliability. When the human teammate enters a building, the true state of any threat within is revealed. At this point, the robot will know which of its sensor readings were correct and which not. For example, consider a false negative where the robot's camera (and image-processing system) fail to detect any hostile occupants, but where its microphone (and natural-language processing system) does detect them. The robot can increase $O(\textit{gunmen}, \textit{camera} = \textit{no gunmen})$ and $O(\textit{gunmen}, \textit{microphone} = \textit{gunmen})$. As a result, the robot will be more likely to recommend protective gear in future buildings when its microphone is positive for gunmen and its camera is negative.

Model-based RL can update the parameters of the robot's POMDP model in this way. It can then use this now-dynamic POMDP model to feed the same explanation generation used for the static POMDP models in our past work. In particular, if the agent is updating its $O$ function, then an $O$ template (as described in Section 3.2) could make the revision transparent by saying "My image processing will fail to detect armed gunmen 45% of the time.". The teammate will thus be informed as to the changes in $O$ resulting from the model-based RL.

### 4.2 Model-Free RL

Alternatively, the agent can use the POMDP model to compute an initial value function, but then use model-free RL to update those values. In other words, the robot maintains only a set of Q values, $Q(b, a)$, throwing away the POMDP model after initializing them. In our particular domain, the agent's belief state, $b$, is determined by its sensor readings in the current building. We can thus represent $b$ as a tuple $\langle c, n, m \rangle$ of the robot's camera, NBC sensor, and microphone readings, respectively.

To initialize $Q(\langle c, n, m \rangle, a)$, we first solve the robot's prior POMDP model to arrive at a value function, $V$, over belief states, $b$, and actions, $a$. We can then iterate through each combination of sensor readings, $\langle c, n, m \rangle$, compute the belief state derived from those sensor readings, $b$, and then assign the corresponding value to our $Q$ values: $Q(\langle c, n, m \rangle, a) \leftarrow V(b, a)$.

We can then follow a standard model-free RL formulation to update these $Q$ values based on the robot's experiences. The robot will receive a reward signal after its recommendation, based on whether the human teammate suffered any injury and whether time was lost from putting on protective gear. This reward signal will be used to update the $Q$ values for the current sensor readings. Over time, these $Q$ values should approach the optimal values for recommending protective gear based on the sensor readings of threats in the building.

It is important to note that these $Q$ values will account for the effect of broken sensors, disobedient teammates, etc., but with no explicit representation of these factors. For example, if the robot's camera is broken so that it never detects gunmen, the $Q$ values will converge to a table whose values are change based on only the NBC sensor on microphone. However, there will be no explicit representation of the observation function, $O$, as there was for the model-free case.

## 5 Explanation Generation for RL

Enabling the agent to learn should enhance its domain-level performance and potentially improve team performance. However, the unpredictability introduced by allowing the agent to change its model (and its decision-making) can lead to distrust from human teammates. It therefore becomes important for the agent to be able to make its learning transparent, as well as its decision-making.

### 5.1 POMDP-Based Explanation

We first seek to leverage our existing POMDP-based explanation templates. When our robot is using model-based RL, it is trivial to apply the templates from Section 3.2. At each stage, the robot has a POMDP model of the world, and it uses that POMDP to make its decisions. Therefore, any learned components (such as the $O$ example, given in Section 4.1) can be directly fed into the POMDP templates to reveal the agent's current model. Of course, the RL agent's POMDP model will change over time, causing these explanations to change over time as well. However, the dynamics of these explanations can potentially reveal enough information for the teammates' to understand why the robot behavior has changed.

On the other hand, model-free RL (as described in Section 4.2) does not maintain a POMDP model, but rather just the $Q$ values that specify the values associated with all possible actions in a given belief state. Therefore, we cannot directly apply our POMDP explanation templates. However, unless our POMDP modeling structure is erroneous, then there are likely to be many POMDPs that are consistent with the policy arrived at by model-free RL. Therefore, if we can find a POMDP that matches the learned policy, then we can potentially use it as the input to our POMDP explanation templates, just as we can for model-based RL.

To find such matching POMDPs, we first consider a set of candidate matches. One such set is the set of possible POMDPs that can be learned by our model-based RL alternative (as described in Section 4.1. In our HRI domain, model-based RL can vary only $O$, so we need to consider only the variations of our initial POMDP that differ in $O$. We can discretize the space of possible $O$ functions to arrive at a finite set of candidate POMDPs.

For each such candidate POMDP, we can compute the optimal policy for eventual comparison against the learned policy. We can potentially perform this computation using any existing solver, but we exploit an assumption of piecewise-linear models [Pynadath and Marsella, 2004] to arrive at a decision-tree representation of the optimal policy for both our candidate POMDPs and the learned $Q$ values. We simply look for a POMDP whose decision-tree policy matches the decision-tree policy of the learned $Q$ function. We then use that POMDP as the input to Section 3.2's templates, just as we did for the model-based RL's POMDP.

It is likely that there are multiple POMDPs consistent with the learned $Q$ values, because although parameter changes will change the POMDP value function, most will not change the optimal policy. In such cases, we could simply pick one of the consistent POMDPs. It is an empirical question of how best to make this choice so that the agent provides explanation content that best calibrates trust with its human teammate.

Alternatively, we could focus on only the modeling components that overlap among the matching POMDPs. For example, if all the matching POMDPs have the same $P$ function, but different $O$ functions, then we would ignore $P$ explanations and focus on $O$ ones instead. Of course, it is possible that the ambiguity among the matching POMDPs extends to the individual components, in which case this method offers no advantage.

### 5.2 Learning-Based Explanation

The POMDP-based explanation from Section 3.2 reveals individual components of the agent's learned model. However, it does not inform the human teammate as to the overall content learned, nor how the learning arrived at that content. Fortunately, we can create additional templates for the components introduced by enabling our agent to use RL to update its model.

**$Q$:** The agent can explain its decisions by communicating the $Q$ values that led to them. For example, the robot could say that "I currently estimate that not wearing protective gear is over 3 times better than wearing it."

**$Q(b, \cdot)$:** Alternatively, it could include the belief state on which the $Q$ values are currently conditioned. For example, "Not wearing protective gear is over 3 times better than wearing it when there is a 90% chance that there is no threat."

**$Q(\vec{\omega}, \cdot)$:** As another alternative, the agent could describe its belief state in terms of the observations from which its current belief state is derived. For example, "Not wearing protective gear is over 3 times better than wearing it when none of my sensors detect any threats."

**$\pi(b)$:** While providing the $Q$ values in the current belief state certainly helps transparency, the teammate might benefit even more from understanding the agent's overall decision-making policy. For example, the robot could communicate context about the above examples by saying, "I recommend not wearing protective gear when I believe that there is a less than 25% chance of a threat." We leverage our piecewise-linear assumption to be able to generate such decision-tree representations of the agent's policy [Pynadath and Marsella, 2004].

**$\pi(\vec{\omega})$:** We can again formulate an alternative policy-based explanation using the sensor readings on which the relevant policy entry is contingent. For example, "I recommend not wearing protective gear whenever neither my NBC sensor nor microphone detect a threat."

**Data:** The $Q$ values and policy certainly increase transparency, but they do not inform human teammates as how they were derived. It might therefore be useful to use the data used in the RL to arrive at the agent's current $Q$ values. For example, "I recommend not wearing protective gear, because only 3 times (out of 91) was there actually a threat when none of my sensors did not detect any."

## 6 Discussion and Future Work

Just as with the POMDP-based explanation components of Section 3.2, it is an empirical question as to the impact of these different explanation forms on human-robot trust. The necessary next step is to conduct human-subject studies with our learning agent across different permutations of these explanations. Such an evaluation of the RL-based explanation can not only inform the efficacy of such explanations on building transparency, calibrating trust, and repairing trust, but also provide insight into what aspects of the RL should be included in the explanations and how to present them. For example, high-level explanations can provide at-a-glance information to help make decisions quickly, while detailed explanations can better help human teammates understand the agent's decision-making process. Human-interaction data can help evaluate such trade-offs and enable future agents to choose the right type of information to communicate at the right level of detail. More importantly, such evaluations can potentially reveal mechanisms to adapt to a human teammate's changing information and decision needs.

Another extension of the current work is to design bi-directional transparency communication that not only allows an agent to communicate its decision and explanations to its human teammate, but that also supports human input to the agent's decision-making, including interactive reinforcement learning similar to [Fukuchi *et al.*, 2017]. Such an investigation can also assist in more general classification of the individual differences that are prevalent in HRI domains [Pynadath *et al.*, 2018].

This paper's RL-based explanation mechanisms provide a simple way to systematically explore the impact of different XAI content on human trust perceptions and behavior in an HRI domain. Although most RL domains are more complex than our testbed domain, the POMDP and Q-learning components are identical. Therefore, we are hopeful that our findings about how different content affects human teammates will generalize to more realistic domains as well. As such, this paper outlines a potentially fruitful line of investigation into how best to automatically generate explanations of RL to benefit human-agent team performance.

## Acknowledgments

This project is funded by the U.S. Army Research Laboratory. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

## References

[Boutilier *et al.*, 2000] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial Intelligence*, 121(1):49–107, 2000.

[Dzindolet *et al.*, 2003] Mary T. Dzindolet, Scott A. Peterson, Regina A. Pomranky, Linda G. Pierce, and Hall P. Beck. The role of trust in automation reliance. *International Journal of Human-Computer Studies*, 58(6):697–718, 2003.

[Elizalde *et al.*, 2008] Francisco Elizalde, L. Enrique Sucar, Manuel Luque, J. Diez, and Alberto Reyes. Policy explanation in factored markov decision processes. In *Proceedings of the European Workshop on Probabilistic Graphical Models*, pages 97–104, 2008.

[Fukuchi *et al.*, 2017] Yosuke Fukuchi, Masahiko Osawa, Hiroshi Yamakawa, and Michita Imai. Autonomous self-explanation of behavior for interactive reinforcement learning agents. In *Proceedings of the 5th International Conference on Human Agent Interaction*, pages 97–101. ACM, 2017.

[Hayes and Shah, 2017] Bradley Hayes and Julie A Shah. Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the 2017 International Conference on Human-Robot Interaction*, pages 303–312. ACM, 2017.

[Hendricks *et al.*, 2016] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016.

[Iyer *et al.*, 2018] Rahul Iyer, Yuezhang Li, Huao Li, Michael Lewis, Ramitha Sundar, and Katia Sycara. Transparency and explanation in deep reinforcement learning neural networks. In *Proceedings of the Conference on Artificial Intelligence, Ethics, and Society*, 2018.

[Kaelbling *et al.*, 1996] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[Kober *et al.*, 2013] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[Lewicki, 2006] R. J. Lewicki. Trust, trust development, and trust repair. In M. Deutsch, P. T. Coleman, and E. C. Marcus, editors, *The handbook of conflict resolution: Theory and practice*, pages 92–119. Wiley Publishing, 2006.

[Matarić, 1997] Maja J. Matarić. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83, 1997.

[Parasuraman and Riley, 1997] Raja Parasuraman and Victor Riley. Humans and automation: Use, misuse, disuse, abuse. *Human factors*, 39(2):230–253, 1997.

[Pynadath and Marsella, 2004] David V. Pynadath and Stacy C. Marsella. Fitting and compilation of multiagent models through piecewise linear functions. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 1197–1204, 2004.

[Pynadath *et al.*, 2018] David V. Pynadath, Ning Wang, Ericka Rovira, and Michael J. Barnes. Clustering behavior to recognize subjective beliefs in human-agent teams. In *International Conference on Autonomous Agents and Multiagent Systems*, 2018.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[Robinette *et al.*, 2015] Paul Robinette, Ayanna M. Howard, and Alan R. Wagner. Timing is key for robot trust repair. In *International Conference on Social Robotics*, pages 574–583. Springer, 2015.

[Russell and Norvig, 2016] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[Schweitzer *et al.*, 2006] Maurice E. Schweitzer, John C. Hershey, and Eric T. Bradlow. Promises and lies: Restoring violated trust. *Organizational behavior and human decision processes*, 101(1):1–19, 2006.

[Sheh, 2017] Raymond Sheh. " why did you do that?" explainable intelligent robots. In *AAAI Workshop-Technical Report*, pages 628–634, 2017.

[Smart and Kaelbling, 2002] William D. Smart and Leslie Pack Kaelbling. Effective reinforcement learning for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3404–3410. IEEE, 2002.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[Swartout and Moore, 1993] William R. Swartout and Johanna D. Moore. Explanation in second generation expert systems. In *Second generation expert systems*, pages 543–585. Springer, 1993.

[Swartout *et al.*, 1991] William Swartout, Cecile Paris, and Johanna Moore. Explanations in knowledge systems: Design for explainable expert systems. *IEEE Expert*, 6(3):58–64, 1991.

[Wang *et al.*, 2015] Ning Wang, David V. Pynadath, and Susan G. Hill. Building trust in a human-robot team. In *Interservice/Industry Training, Simulation and Education Conference*, 2015.

[Wang *et al.*, 2016] Ning Wang, David V. Pynadath, and Susan G. Hill. The impact of POMDP-generated explanations on trust and performance in human-robot teams. In *International Conference on Autonomous Agents and Multiagent Systems*, 2016.

[Wang *et al.*, 2018] Ning Wang, David V Pynadath, Ericka Rovira, Michael J Barnes, and Susan G Hill. Is it my looks? or something i said? the impact of explanations, embodiment, and expectations on trust and performance in human-robot teams. In *International Conference on Persuasive Technology*, pages 56–69. Springer, 2018.

[Zhang *et al.*, 2017] Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. Plan explicability and predictability for robot task planning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1313–1320. IEEE, 2017.