Evaluating the Role of Interactivity on Improving Transparency in Autonomous Agents

Peizhu Qian
Rice University
Houston, TX, USA
pqian@rice.edu

Vaibhav Unhelkar
Rice University
Houston, TX, USA
vaibhav.unhelkar@rice.edu

ABSTRACT

Autonomous agents are increasingly being deployed amongst human end-users. Yet, human users often have little knowledge of how these agents work or what they will do next. This lack of transparency has already resulted in unintended consequences during AI use: a concerning trend which is projected to increase with the proliferation of autonomous agents. To curb this trend and ensure safe use of AI, assisting users in establishing an accurate understanding of agents that they work with is essential. In this work, we present AI Teacher, a user-centered Explainable AI framework to address this need for autonomous agents that follow a Markovian policy. Our framework first computes salient instructions of agent behavior by estimating a user's mental model and utilizing algorithms for sequential decision-making. Next, in contrast to existing solutions, these instructions are presented interactively to the end-users, thereby enabling a personalized approach to improving AI transparency. We evaluate our framework, with emphasis on its interactive features, through experiments with human participants. The experiment results suggest that, relative to non-interactive approaches, interactive teaching can both reduce the amount of time it takes for humans to create accurate mental models of these agents and is subjectively preferred by human users.

CCS CONCEPTS

• **Computing methodologies** → **Intelligent agents**; *Markov decision processes*; *Theory of mind*; *Planning under uncertainty*; • **Computer systems organization** → **Robotics**; • **Human-centered computing**;

KEYWORDS

Explainable AI; Machine Teaching; Human-AI Collaboration; Shared Mental Models; Monte-Carlo Tree Search

ACM Reference Format:
Peizhu Qian and Vaibhav Unhelkar. 2022. Evaluating the Role of Interactivity on Improving Transparency in Autonomous Agents. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 9 pages.

1 INTRODUCTION

Autonomous agents provide services to a broad array of end-users in homes, offices, hospitals, and beyond. They are expected to collaborate with people on complex tasks such as disaster response, manufacturing, and agriculture [5, 8, 17, 43]. While these agents behave according to policies carefully designed by their programmers, their resulting behavior may not always be intuitive to end-users. For example, consider a robot designed to support first responders in rescue operations. To best use and work with the robot, a first responder might wonder how the robot plans its route to rescue victims: will it *go through* uneven terrain if that provides the shortest path, or will it *circumvent* said terrain to avoid the risk of accidents? Knowing the answers to questions such as these is essential for human end-users to truly realize the benefits of these agents, ensure safety, and avoid unintended side effects [20, 32].

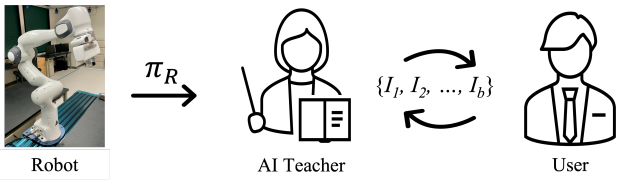

Figure 1: Our goal is to assist users in establishing accurate mental models of robot behavior (π_R) by designing an AI Teacher that interactively provides instructions $\{I_{1:b}\}$.

However, answering such questions can be difficult for end-users. One reason for this difficulty is that agent[1] behavior is often designed to optimize a latent objective (e.g., time to complete a task), and not to ensure transparency in the generated behavior. For instance, the algorithmic paradigms used for designing behavior (e.g., reinforcement learning [39] and planning under uncertainty [13, 33]) seek to maximize the cumulative reward, which typically omits any consideration of AI transparency. Despite this, humans create mental models of agents they work with and use them to make inferences and predictions [19, 27]. Unfortunately, without appropriate interventions, the process of creating these models can be slow and inaccurate, leading to AI misuse or disuse [12, 23, 32].

Thus, to ensure safe and appropriate use of autonomous agents, the onus of establishing accurate mental models regarding their behavior lies with us: AI researchers and designers. Towards this call to action, we seek to create an interactive AI Teacher that can assist end-users in establishing accurate understanding of the behavior of autonomous agents (see Fig. 1). In this work, we focus on agents that are tasked with performing sequential tasks and whose behavior can be summarized using a Markovian policy. Recognizing the need for AI transparency, there is a growing body of research on explaining AI systems to humans. Most of the work in this burgeoning area has explored approaches to explaining the

[1]We use the terms *robot* and *agent* interchangeably.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

predictions of machine learning models [10, 29, 42]. Closer to our focus, multiple formative approaches to explain sequential decision-making behavior of robots have also been developed in recent years [21, 38, 40, 44]. Broadly, these approaches computationally generate human-interpretable instructions (typically, explanations or examples) to communicate robot objectives.

While these approaches further AI transparency and have inspired our work, borrowing terminology from pedagogy [24, 30, 34], they teach humans about the agent by providing either direct instructions or indirect instructions *but not both*. In approaches that utilize direct instructions, the teaching algorithm precomputes a set of informative instructions, typically without any personalization, which are then presented to the user [21, 38]. Drawing connections to pedagogy, this setting is analogous to that of the traditional classroom, where the teacher decides the teaching material and delivers it to the students while approaches that utilize indirect teaching provide the users a tool to ask questions and expect them to learn from exploration analogous to active learning [14, 18, 26]. In contrast, effective human-to-human teaching involves a **hybrid** of direct and indirect instructions to allow for effective and personalized teaching [34]. Direct instructions ensure that the salient information is conveyed, while indirect instructions allow for the students to personalize their learning.

Guided by this, we posit that the use of such hybrid strategies can also benefit the realization of AI transparency. To evaluate this hypothesis, we make **two core contributions** in this paper. First, we develop AI Teacher, a human-in-the-loop Explainable AI framework to communicate robot behavior to human users. Our framework computes direct instructions using an algorithm for sequential decision-making and guides users to learn through indirect instructions with an interactive user interface. Second, we provide a detailed human subject evaluation of our hybrid approach by applying it on three tasks and benchmarking it against the direct and indirect paradigms to AI transparency. To the best of our knowledge, these experiments are the first to evaluate the effect of interactivity on AI transparency and shed light on the utility of interactivity on improving user's understanding of autonomous agents. Experiment results show that a hybrid approach outperforms direct instructions in improving user knowledge. The use of interactivity features enable users to verify individual understanding of robot policy and objectives while keeping users more engaged during the learning process. Guided by these results, we conclude with recommendations for design of computational techniques and interactive systems to improve AI transparency.

2 PRELIMINARIES

Representing Agent Behavior. In the following treatment, we focus on communicating task-oriented behavior of agents, where the agent's task can be modeled as a Markov decision process (MDP) [33]. Briefly, an MDP is specified using the tuple $M \doteq (S, A, T, R, \gamma)$, where $s \in S$ is the set of states; $a \in A$ is the set of actions available to the agent; $T(s'|s, a)$ is the Markovian state transition probabilities; $R(s, a)$ the reward function; and γ the discount factor. The agent has full observability of the state and acts according to the policy $\pi_R(s) = a$. This modeling choice is similar to that made in related work on explaining and summarizing robot behavior [1, 2, 21, 28].

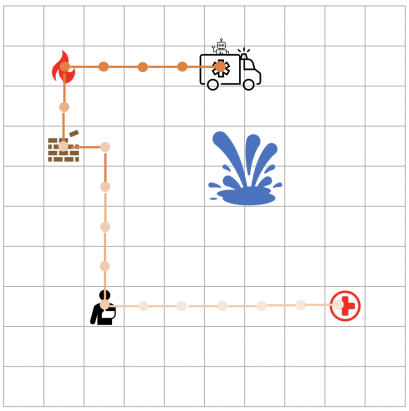

Figure 2: The simulated Rescue robot and its environment.

While we limit our scope to agents whose behavior can be summarized by the Markovian policy π_R, no assumptions are made regarding the procedure to compute this policy. This is important in practice, where the agent policy can be hand-crafted, planned [33], or learned [39] and, thus, need not to be optimal.

Running Example. Inspired by applications, we design a disaster response scenario where an autonomous robot is performing rescue operations. We utilize this disaster response robot, shown in Fig. 2, as a running example to elaborate on our approach and design decisions. The robot's environment is a grid world with various landmarks. The robot can fully observe its environment and where it can take one of the following four navigation actions: move up, move down, move left, or move right. The robot is preprogrammed to avoid the pond and complete available sub-tasks based on the following priority: extinguish the fire, clean up the debris, pick up the patient, and visit the hospital. A sub-task is completed when the robot reaches the corresponding grid location. Our research goal is to provide end-users (e.g., a first responder working with the robot) accurate understanding of the robot behavior.

For this example, the robot's task is represented as an MDP, with state s including features corresponding to agent location and status of each sub-task. Similarly, the action space A corresponds to the four navigation actions. The robot behavior is summarized using a Markovian policy π_R, which maps its state to action. With the awareness of robot policy, we posit that the user will be able to make informed predictions of robot behavior both at the task (e.g., will the robot extinguish the fire first or pick up the patient?) and motion level (e.g, which direction will the robot move next?). This will enable the user to work with the robot effectively both in unseen and emergent situations.

3 RELATED WORK

We begin with a brief review of related work on improving transparency in behavior of autonomous agents. Interspersed with this review, we also draw connections between solutions for improving transparency to the rich literature on pedagogy.

The problem of improving transparency in agent behavior is related to the emerging research areas of AI explainablility, interpretability, and transparency. For a detailed review of these areas in the context of autonomous agents, we refer the reader to recent surveys by Anjomshoae et al. [3] and Chakraborti et al. [15]. Among the various threads being actively explored in these emerging areas, our work on transparency is related to methods for effectively communicating agents' objectives and sequential policies to humans [2]. In our review, we focus on these methods, with an emphasis on those that model agent behavior as a Markovian policy.

Researchers in pedagogy have examined a variety of teaching strategies for the setting of human-to-human teaching [24, 30, 34]. One categorization of these strategies is based on the use of direct versus indirect instructions. Teaching strategies utilizing direct instructions are teacher-centered, involve clear teaching objectives, and consistent classroom organizations. In contrast, strategies involving indirect instructions are student-centered and encourage independent learning. Effective teaching typically involves a combination of the two perspectives. The process of improving transparency in AI agents can be viewed from the lens of pedagogy, wherein the algorithm for improving transparency is the *teacher*, the human user corresponds to the *student*, and the learning objective reflects an accurate understanding of the *agent* behavior.

3.1 AI Transparency via Direct Instructions

We first describe methods that utilize direct instructions. A common theme in these methods is to precompute informative instructions regarding agent behavior which are then communicated to the human. The methods differ in their choice of instruction type, objective criteria, and optimization algorithm.

Instruction Type. Information about the agent can be delivered to the human using a variety of modalities (e.g., text, images, augmented/virtual reality) and types (e.g., natural language explanations, template-based explanations, demonstrations). To arrive at methods that can be applied across domains, instruction types such as policy summaries, important states, and key actions have been previously explored [21, 22, 25, 28, 38, 41, 44]. Among these, our work utilizes examples (i.e., state, action-pairs) of agent behavior as the domain-agnostic instruction, $I \doteq (s, a)$.

Selecting Informative Instructions. Having identified the instruction type, a teaching algorithm needs a mechanism to select *informative* instructions from the set of possible instructions. This is achieved by specifying an objective criteria to quantify informativeness of instructions, which is then used within an optimization routine to select the examples to be shared with the user. Multiple alternatives have been previously explored for the case of examples as the instruction type, both with and without explicit user modeling. For instance, Zhan et al. [44] quantify informativeness using the importance value, $\textsc{Importance}(s) = \max_a Q(s,a) - \min_a Q(s,a)$, where $Q(s,a)$ denotes the expected cumulative reward of a ⟨state, action⟩ pair. Their teaching algorithm shares the instructions with the highest importance value with the user. Amir and Amir [1] also utilize this objective and augment it to ensure diversity in the generated instructions.

Towards Personalized Instructions. We highlight that Importance criterion depends only on the task (MDP) and agent policy, thereby leading to identical instructions for each user. A precursor for personalized teaching is an objective criteria that depends on the user. Towards this, Huang et al. [21] provide an approach that explicitly represents a user's mental model of the agent behavior as a probability distribution over candidate agent rewards. By utilizing Bayesian Theory of Mind [6], they model the effect that each instruction I has on the user's understanding of the agent behavior. The instruction that improves the user understanding the most based on this Bayesian model is said to be the most informative and selected using a greedy algorithm. Lee et al. [28] provide a related approach where the user is modeled as an inverse reinforcement learner. While these methods model user knowledge, they rely on direct instructions alone and, thus, have limited freedom for personalizing instructions.

3.2 AI Transparency via Indirect Instructions

An alternate paradigm for improving AI transparency is to provide humans the tools to ask questions regarding the agent's behavior [14, 18, 26]. Analogous to indirect teaching methods in pedagogy, this method leaves it to the student (human user) to identify which information she considers to be the most informative for her learning experience. Thus, instead of identifying the most informative instructions, the computational contribution of these methods lies in effectively answering the questions posed by the user. For instance, Hayes and Shah [18] provide a tractable method to answer the question, "in which states does the agent perform a queried action?" While focusing on deterministic plans instead of MDP policies, Chakraborti et al. [15] discuss several other question types that have been recently considered.

Assuming that the student knows what she does not know, this approach can be indeed powerful. However, this assumption of *unknown unknowns* may not be generally applicable. Hence, we explore the use of hybrid teaching strategies. Our insight is that *methods for generating direct instructions can help provide the user with adequate knowledge about the agent, which the user can refine and improve upon using the methods for indirect instructions.* Guided by this insight, we design an interactive AI Teacher, which to our knowledge is the first framework to employ both direct and indirect instructions to improve transparency in agent behavior.

4 PROBLEM STATEMENT

Consider an agent performing a sequential task, modeled as an MDP $= (S, A, T, R, \gamma)$, based on policy π_R. Collocated with the agent is a human user, who has complete knowledge of the task environment (S, A, T) but only an estimate π_U of the agent's policy. We denote the user's knowledge of the policy as

$$K_U = \frac{1}{|S|} \sum_{s \in S} \mathbb{1}\left[\pi_R(s) = \pi_U(s)\right] \tag{1}$$

which describes the fraction of states s for which the user's policy estimate is correct. $K_U = 0$ means the user has no knowledge, while $K_U = 1$ denotes an accurate mental model of the agent behavior. The user can improve her estimate of the agent policy by requesting direct and indirect instructions. Each instruction I corresponds to a

(s_I, a_I)-pair, where the action $a_I = \pi_R(s_I)$. Direct instructions are generated by the AI Teacher, while the indirect instructions are the teacher's response to user's question "Which action does the agent take in state?" Formally, the objective of the AI Teacher is to generate a sequence of b direct instructions to maximize the user knowledge, $\text{argmax}_{I_{1:b}} K_U$.

Solving this optimization problem is challenging for a variety of reasons. First, the AI Teacher needs a mechanism to reason about the effect of an instruction on user's knowledge, a latent quantity that depends on user's expertise and learning preferences. Second, the AI Teacher needs a computationally tractable approach to identify not only the most informative instructions but also their sequence, as some instructions might be more effective initially. Lastly, the user can interleave direct instructions with independent learning (indirect instructions), making the problem of selecting effective instruction further challenging.

5 TECHNICAL APPROACH: AI TEACHER

To tackle the problem described in Sec. 4, we design the framework AI Teacher that includes a model of human cognition, an algorithm to select instructions, and an interactive user interface for teaching. We mathematically represent the user's mental model about the agent policy using Griffith's probabilistic model of cognition [16]. The effect of instructions on the evolution of mental model is captured using Bayesian Theory of Mind [6]. Given this representation, we pose the problem of selecting instructions to maximize user knowledge as a sequential decision-making problem and solve it using Monte Carlo tree search (MCTS) [11]. Finally, our framework includes an interactive user interface that allows a user to self-explore the agent behavior akin to independent learning.

5.1 User Model

To select effective and personalized instructions, AI Teacher needs a reliable estimate of the human user's mental model and learning process. However, modeling these cognitive quantities is challenging as they depend on latent and user-specific features, such as prior knowledge, experience, and attention during the teaching process. In our framework, we leverage results from cognitive science to arrive at these models for the human user [7, 9].

Mental Model. Theory of Mind (ToM) refers to the human ability to interpret and predict behavior of others by attributing mental states (e.g., belief, desire, and intent) to oneself or others [7, 9]. Recent work indicates that humans also attribute ToM to artificial agents that they observe or interact with [19, 27]. Guided by these results, to quantify the user's mental model, we model the user to have a set of candidate hypotheses E about the agent. Each hypothesis $e \in E$ corresponds to an estimate π_e of the true agent policy π_R. For instance, in our running example (Fig. 2), a user might have the following candidate hypotheses: (i) the robot will complete the closest task first, and (ii) the robot will pick up the patient first because it is driving an ambulance. The user's mental model of the policy is represented as a probability distribution over all candidates, referred to as belief b. The belief $b(e)$ denotes the probability of the user believing π_e to be the correct estimate of the true robot policy. At the start of the teaching process, $b(e)$ summarizes user's prior knowledge about the agent policy.

Effect of Instructions. Given this mental model, we next utilize Bayesian Theory of Mind to estimate the effect of an instruction I on the user's belief b. Mathematically, the updated belief b' after receiving an instruction I is computed using the Bayes' rule,

$$b'(e) = \Pr(e|I) \propto \Pr(I|e)b(e) \tag{2}$$

where, $\Pr(I|e)$ models the likelihood of the user expecting to receive instruction I from the teacher given π_e is the correct estimate. In our implementation, we model this likelihood as $\Pr(I|e) = \pi_e(a_I|s_I)$. Intuitively, this model prunes those hypotheses from the candidate set that do not align with the instructions.

Hypothesis Set. A key parameter in our user model is the set of hypotheses E. To accurately model users, it is important to specify the possible hypotheses they might have about the agent, a challenging task due to the diverse experience and background of AI users. Nonetheless, to approximate this parameter in practice, we discuss three methods each with different pros and cons. First, when users can be surveyed before the teaching process, the set E can be designed using their domain expertise. Hypotheses collected using this method will reflect the diversity in users but can be difficult to translate to mathematical representation. Second, in cases where users can provide demonstrations of agent's expected behavior (e.g., through teleoperating a robot), the hypotheses can be approximated using algorithms for imitation learning [4]. In practice, this approach may require prohibitive amount of data; however, with continued advances in imitation learning, we anticipate this approach to be more data-efficient [31]. Last, if users can be neither surveyed nor provide demonstrations, hypotheses can be generated through sampling potential agent reward functions.

We utilize the third approach in our experiments. For an MDP task, with reward given as a linear combination of features ($R = \sum_i w_i f_{s,a}$), we first compute a set of candidate hypotheses over agent rewards as permutations over the feature weights,

$$E = \left\{ \sum_i v_i f_{s,a} \mid v_i \in \text{permutation}(w) \right\}. \tag{3}$$

The policy estimate π_e corresponding to each candidate reward is then precomputed using MDP solvers, such as policy iteration or reinforcement learning. We reiterate that modeling cognition is inherently challenging and that our approach is but one alternative to arrive at a tractable approximation of the user's learning process. To assess the robustness of AI Teacher to this modeling choice, we conduct both synthetic and human experiments.[2]

5.2 Instruction Selection

We pose the problem of selecting the optimal sequence of instructions as a sequential decision-making problem. To do so, we view the AI Teacher as the decision-maker, who can use its instructions to improve user knowledge K_U. Specifically, we define a Teaching MDP, whose state corresponds to the user belief b and action corresponds to the teacher's instruction I.[3] The state space for this

[2] Estimation techniques for users' hypothesis set E and likelihood model $\Pr(I|e)$ remains a fruitful avenue for future research. We design our instruction selection procedure to be modular, so that it can be readily used with other user models.

[3] As user belief is a latent quantity, this model can be extended to consider partial observability and information gathering actions (such as AI Teacher asking questions to the user). We leave exploration of these extensions to future work and, in this work, proceed with the assumption of state observability for tractable planning.

MDP is the continuous set of all possible beliefs. The action space is the set of all instructions I, whose size is equal to the number of states in the Task MDP described in Sec. 4. The solution of the Teaching MDP corresponds to the teaching policy, which maps user belief to teacher's instructions. To complete the specification of the Teaching MDP, transition and reward models are needed. Given the user model, the transition model is readily available from Eq. 2, which provides the updated belief (next state in the Teaching MDP) based on the previous belief (state) and instruction (action). We define the reward function of the Teaching MDP to reflect the problem objective of maximizing user knowledge as the belief-weighted estimate of a user's knowledge K_U,[4]

$$R_{\text{teacher}}(b', I, b) = \sum_{e \in E} b'(e) K_e \tag{4}$$

Having specified the Teaching MDP, we can use MDP solvers to arrive at the teaching policy. However, as the Teaching MDP involves a continuous state space and (potentially) large action space, computationally efficient methods are needed. Thus, we utilize Monte Carlo tree search (MCTS) for MDP [11], inspired by its wide use in applications which require fast, online sequential decision-making and include high branching factors [35, 36]. In contrast to prior work [21], our approach allows for selecting the instruction sequence in a non-greedy fashion and for stochastic domains. Here, we describe the essential details for using MCTS to select instructions. For a detailed description of MCTS, we refer the reader to the survey by Browne et al. [11].

Briefly, MCTS involves creation of a tree using the following steps: selection, expansion, simulation, and backpropagation. In our application, the root node of the MCTS tree represents the prior belief of a user, for instance, a uniform distribution over candidate hypotheses. Each edge represents an instruction $I = (s, a)$ and each child node consists of the updated belief $b'(e)$. The output of the MCTS algorithm is the best instruction for a given belief. In practice, it is often easier to demonstrate agent trajectories instead of disjoint state, action-pairs. This feature can be readily incorporated in MCTS, via its simulation step, by constraining successive instructions to correspond to a trajectory. In our implementation, we use trajectories of length 5 to generate instructions; drawing connection to the pedagogy, our user interface refers to each trajectory (sequence of 5 instructions) as a lesson. Thus, in combination with the user model, the MCTS algorithm method generates sequence of direct instructions to improve knowledge of the (modeled) user.

5.3 Avenues of Interactivity

In addition to the direct instructions computed as described above, the AI Teacher includes interactive features to allow for independent learning: namely, indirect instructions and quiz problems. When interacting with the AI Teacher, a user can design a subset of the instructions. Specifically, the user can request context-specific examples of agent behavior by asking the question, "which action does the agent take in state?" This feature is achieved through an interactive user interface, which allows users to modify the state of the task (e.g., robot position and environment features). We refer to these customizable indirect instructions as **custom instructions**. These instructions help users verify user-specific assumptions or theories about the agent policy and enable our approach to be student-centered to the extent desired by the user.

Further, to facilitate engagement, AI Teacher also gives the user quizzes according to a predefined schedule. In the quiz problems, AI Teacher presents a state and asks the user to predict the agent's action in the given state. The quiz problems can be selected in a variety of methods, e.g., randomly, based on user's learning thus far, or using the MCTS algorithm. In our framework, we use the first state of each lesson selected by the MCTS algorithm. For purposes of fair evaluation against baselines, we do not provide feedback to the user for their response. However, the users can utilize the mechanism for custom instructions to check if their response is accurate.

6 EXPERIMENTAL DOMAINS

We design three simulated domains inspired by a different real-life application to evaluate our framework: autonomous navigation, disaster response, and recycling. For generating direct instructions in each domain, we approximate the free parameter E of the user model as described in Sec. 5.1. While the environments chosen in our experiments are simulations of real-world settings, they are challenging and have reasonably large state spaces. In contrast to domains used in prior work [21, 28], which involve deterministic state transitions, our work also considers stochastic environments and dynamic elements, e.g., the conveyor belt in the recycling environment. Towards ensuring the generalizability of our framework, our environments are different in their state and action spaces.

Rescue Robot. As the first task, we utilize the disaster response scenario shown in Fig. 2. The robot's environment is specified as a 10×10 grid and includes five landmarks: (fire, debris, patient, hospital, pond). The location of each landmark is fixed, but a subset of them may be absent in each problem instance. The task state s is defined as $(x, y, s_{\text{fire}}, s_{\text{debris}}, s_{\text{patient}}, s_{\text{hospital}}, s_{\text{pond}})$, where (x, y) denote robot position and s_{landmark} is a Boolean variable indicating whether that landmark is present or absent in the scene. The state and action spaces are of size 3,200 and 4, respectively. The robot policy is computed by providing positive rewards for completing sub-tasks in the correct order and a negative reward for entering the pond. Thus, the robot learns to avoid the pond and complete available sub-tasks based on the following priority: extinguish the fire, clean up the debris, pick up the patient, and visit the hospital. A user seeking to understand the robot behavior in this task will need to identify the robot's priority list over sub-tasks and the requirement of avoiding the pond. The user's hypothesis set E is approximated using 25 candidates.

Navigation Robot. The second task models a robot navigating a room that includes five doors and patched areas as shown in Fig. 3. At each time step, the robot can take one of the four actions: move forward, move backward, turn left (counterclockwise) or turn right (clockwise). Like the first domain, the environment is a 10×10 grid. The task state is defined as robot position (x, y) and orientation $\theta \in [0°, 90°, 180°, 270°]$. The state space has a size of 400 and we generate 32 candidates to approximate the user's hypothesis

[4] The choice of reward function is informed by the fact that a user can maintain multiple hypotheses $e \in E$ regarding the robot policy. K_e is user knowledge specific to one hypothesis e. If a user maintains only one hypothesis, then $K_U = K_e$. However, since a user might have multiple hypotheses, we utilize the belief-weighted average.

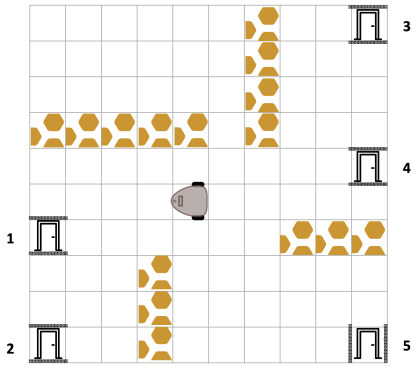

Figure 3: The Navigation robot and its environment.

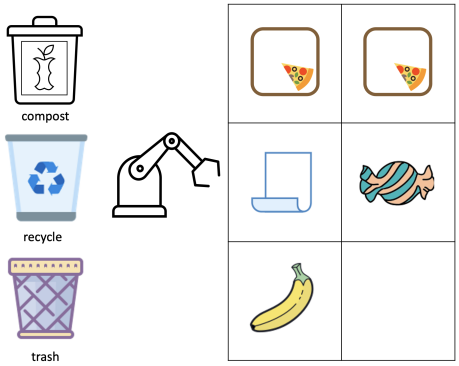

Figure 4: The Recycling robot and its environment.

set, E. The robot's objective (unknown to the user) is to enter the closest odd-numbered door and to avoid the even-numbered doors and the patched areas. Despite its small state space and perceived simplicity, we empirically observe that many human users had the most difficulty understanding the robot behavior in this domain (relative to other two domains, which have larger state spaces). One potential explanation for this is that the robot policy has behaviors which are counter to user's preconceptions. For example, the robot moves backwards, whereas the users might expect it to go forwards.

Recycling Robot. The third task models a recycling robot working in a waste management plan, depicted in Fig. 4. The robot's environment includes three bins (one each for trash, recycling, and compost) and a conveyor belt divided into six regions. The conveyor belts move forward each time step, and new items are introduced in the lowest row. The size of the state space of this task is $\approx$ 80,000, which encodes items on the conveyor belt and with robot. To gain transparency into robot behavior and ensure that it is managing waste correctly, the user seeks to understand which item does the robot pick up from the conveyor belt. To generate the ground truth robot policy, we assign different values to each item and some areas on the conveyor belt are unreachable. For this domain, the user's hypothesis set E is approximated using 24 candidates.

7 EXPERIMENTS WITH SIMULATED USERS

Prior to evaluating with human participants, we validate our approach in Oz-of-Wizard experiments [37]. The simulation studies are an essential pilot experiments to be run before conducting resource-intensive human studies. By simulating the user, we have access to their mental model and can assess the evolution of latent quantities (such as user knowledge). In these experiments, we evaluate the direct instruction component of our system against three algorithms: Importance Advising [44], the approach of Huang et al. [21], and random selection of instructions.

In the interest of space, we defer a detailed discussion of the numerical experiments and results to the supplementary text. Briefly, through these experiments, we observe that our approach outperforms all baselines across the three tasks when AI Teacher has access to the correct user model. In the more challenging case of user model mismatch, our approach is still robust in improving user knowledge. These numerical experiments establish confidence in the ability of our approach to generate effective direct instructions and help us select baselines for evaluations with human users, which are discussed next.

8 EXPERIMENTS WITH HUMAN USERS

We conduct a user study with 24 participants (aged 18 – 31 years, 11 females and 13 males) recruited from Rice University with approval from the Rice University Institutional Review Board (IRB). Through a pre-experiment survey, most participants indicated no prior experience with robots, but some were users of AI systems such as Siri or Google Assistant. The goal of this experiment is two-fold. First, to confirm whether the encouraging results observed with simulated users translate to human users. Second, to discover the role of interactivity (i.e., hybrid teaching strategy) on improving transparency in agent policies.

8.1 Experiment: Design and Procedure

We evaluate our approach (denoted as Hybrid) against two baseline strategies: direct instructions (denoted as Direct) and indirect instructions (denoted as Custom). The approach of Huang et al. [21] is used as the Direct strategy baseline, as it is the baseline that outperformed other baselines in our experiments with simulated users. The Custom strategy corresponds to a variant of our approach without any direct instructions. Specifically, in Custom, the users need to learn the agent policy by requesting custom instructions described in Sec. 5.3. Thus, the experiment has one factor (the teaching strategy) with three levels.

We employ a within-subject repeated measures design. Through an interactive user interface (UI), each participant is tasked with learning the policy of the three robots described in Sec. 6 in the following order: Rescue, Navigation, and Recycle.[5] We vary the order of the teaching strategies to account for any ordering effects. For each of the 6(= 3!) orderings, four participants performed the study. After providing informed consent, the participants are asked to complete a short demographic survey and are briefed about the experiment. To help participants get familiar with the UI, we design a training session where the experiment supervisor (one of the authors) walks the participant through the task.

[5] A video demonstration of the UI is included in the supplementary material.

For each domain and teaching strategy, the participants have a budget of 25 instructions to learn the robot policy. For the Direct strategy, all instructions are generated by the teaching algorithm [21]. For the Custom strategy, all instructions are custom designed by the user by asking questions. For the Hybrid strategy, a participant has the freedom of allocating the budget between direct and custom instructions. For direct instructions, in both Direct and Hybrid settings, we group five consecutive (s, a)-tuple as a trajectory. Once a participant starts a direct instruction, they are required to complete the entire trajectory which will cost them five instructions from the budget. After a participant uses all 25 instructions, they proceed to a post-task subjective survey and an objective exam.

The objective exam is composed of 20 multiple-choice questions, where a participant is presented with a state and asked to predict the robot's action. Half of these questions are sampled randomly from the robot's state space (denoted as Random questions) and the rest are designed by the authors based on our domain expertise of the robot (denoted as Expert Selected questions). The use of both Random and Expert Selected questions allows us to evaluate the user's understanding of the robot policy in a wide sample of states. These questions are followed by one ranking question to evaluate participants' understanding of the robot objectives. Here, the participants are given a list of suggested robot objectives and are asked to rank them. The participants can also remove any of the suggested objectives and add their own explanations of robot behavior. After completing the three domains, the participants complete a post-experiment survey regarding their overall learning experience, perceived role of interactive features, preference over custom and direct instructions, and any open-ended comments.

8.2 Results

Participants learn robot policies despite a small teaching budget. For each robot, the participants score high on both types of the multiple-choice questions regarding robot policy (M = 75%, SD = 21% across tasks and participants) and the ranking questions regarding robot objectives (M = 3.56, SD = 1.67, out of 5 respectively, across all domains and participants). Despite the large state space of the recycling robot (approximately 80,000 states), many participants find it intuitive and are able to quickly build an accurate mental model of the robot's policy and objectives. This result shows that humans have an incredible ability to generalize from a small number of examples. On the other hand, a number of participants find the navigation task to be the most challenging despite its small state space (400 states). This domain is challenging because the navigation robot's behavior is less intuitive to human users (e.g., the robot can move backward). This observation indicates that the complexity of an AI transparency problem does not only depend on the complexity of the state space, action space, or the task itself, but also on the familiarity to users.

Participants' learning experience depends on the teaching strategy. Fig. 5 indicates that participants respond differently to different teaching strategies. To statistically evaluate this effect, we conduct the non-parametric Friedman test. We find a statistically significant effect of teaching strategy on users' subjective perception of the learning experience (p = 0.049). While the effect of teaching strategy on participants' understanding of robot policies

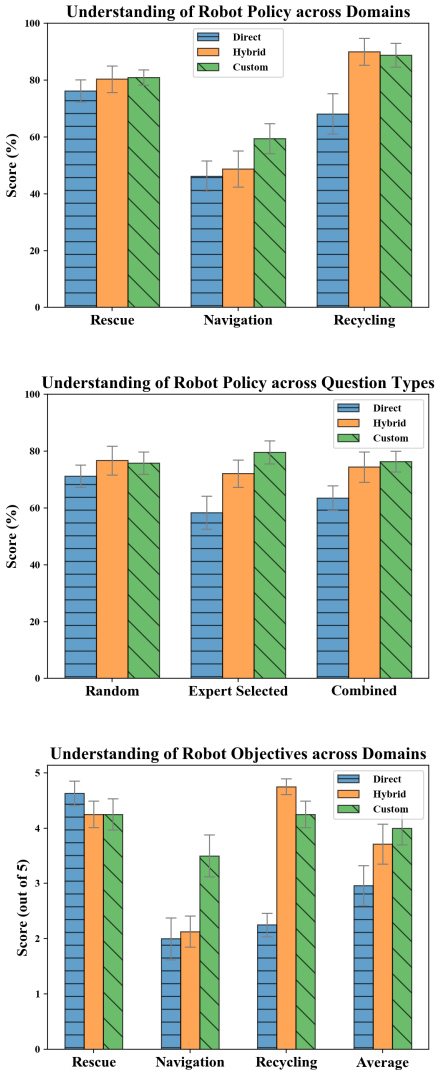

Figure 5: Aggregate results on the post-task objective exams from the experiments with human users.

(p = 0.17) and objectives (p = 0.22) is not statistically significant, potentially due to the sample size (N = 24), the differences on observed learning performance are meaningful and motivate further exploration. We elaborate on these differences next.

Hybrid strategy outperforms Direct Strategy. As shown in Fig. 5 (top, middle), the Hybrid strategy results in higher scores than the Direct strategy in predicting the robot's actions in all domains and across question types. Particularly in the Recycling domain, the Hybrid approach (M = 90%, SD = 13%) improves the participants understanding by 32.1% compared with the Direct approach (M = 68.15%, SD = 20.15%). We posit that the participants utilize direct instructions to form an initial, high-level understanding of the robots, then they utilize custom instructions to clarify corner cases and to verify different hypotheses with a small number of

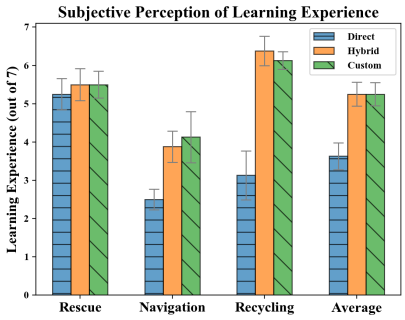

Figure 6: Results from the post-experiment survey.

instructions. As one participant notes in their post-experiment survey, "...the combination of teacher (direct) and custom examples is most useful because it helps you think about new cases and then gives you the tools to test them. I could definitely see how someone could run through the teacher's examples without paying attention if there isn't the interactivity of the custom examples." Subjectively, the participants also prefer Hybrid over Direct approach (Fig. 6).

Hybrid strategy performs comparably to Custom strategy. This trend indicates that human users are capable of identifying critical states that provide the information about the robot that they are previously unaware or unsure of. We posit one reason for the observed performance of the Custom strategy is that the user know *what they do not know* better than the AI Teacher user model can predict and, thus, can choose instructions as effectively as the AI Teacher. In a follow-up study, we plan to compare the overlap between direction and indirect instructions to further analyze human's ability to self-identify critical states as well as to evaluate the accuracy of our human model.

Participants prefer mechanisms for independent learning. The Hybrid and Custom approaches provide the participants an option to learn independently. By creating their own instructions, the participants verify user-specific assumptions. From analyzing the post-experiment survey, we observe a positive trend of utilizing the interactivity features. 92% of the participants rate custom instructions as important (rating ≥ 5 on a Likert-scale of 7) in their learning experience, and 63% of the participants rate quiz problems as important. As one of the participants writes, "[t]he custom examples allowed me to test my hypothesis of what was going on with the priority of tasks. It was hard to figure out the edge cases without being able to use custom examples. The quizzes allowed me to think of new examples and were very useful in combination with the custom examples that allowed me to test these new examples." A number of participants also comment in their feedback that they enjoy the design of the AI Teacher's user interface and making the learning as a game helps significantly with paying attention (particularly from a participant self-identified with ADHD).

Participants prefer the ability to choose between direct and custom instructions. Further, through the survey, three-fourths of the participants indicate that they prefer to (a) receive *both* direct and indirect instructions and (b) choose the relative proportion of each instruction type themselves rather than having an AI teacher to allocate the teaching budget. These responses indicate that participants subjectively prefer receiving both direct and indirect instructions, and further highlight the utility of providing participants both instruction types and the degree of freedom to select between them.

9 CONCLUDING REMARKS

Interactive teaching of agent behavior to humans can reduce the amount of time it takes for humans to create mental models of agents, improve the performance of human-AI teams, and prevent catastrophic consequences caused by poor understanding of AI systems. In this work, we provide AI Teacher, a framework that generates salient examples of agent behavior using a modified Monte Carlo tree search algorithm and includes a user interface to enable teacher-guided exploration of agent behavior. In contrast to prior approaches, AI Teacher provides both direct and indirect instructions to allow for personalized machine teaching. We conduct and report on numerical experiments and a user study to evaluate our approach and the role of interactivity in explaining robot behavior in three Markovian robot tasks. Experiment results show interactivity improves user understanding of robot policies and objectives with a small number of instructions, and methods with interactivity features are highly preferred by human users.

Implication for Designing Explainable and Transparent AI. While existing work focuses on computational methods that generate instructions which help users acquire models of robots, we argue that in absence of interactivity, users are not able to clarify user-specific confusion with only direct instructions. Indirect instructions can address the needs and preferences of individual users. Future XAI systems should leverage on humans' ability of self exploring and independent learning, while providing guidance through computational methods to generate direct instructions. The community can consider diving deeper into studies from education and psychology to design helpful interactivity features.

Future Directions. Our framework offers several avenues for future work. First, we are interested in improving upon the user's cognitive model used by the AI Teacher, by learning its parameters from data and adapting it during the learning process. Second, even though our framework is demonstrated for explaining robot behavior, it is a generalized machine teaching framework. Hence, we plan to explore performance of our approach on other machine teaching tasks. Last, we are interested in incorporating personalization during the generation of direct instructions through information gathering actions.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their detailed and constructive feedback. Peizhu Qian was partially supported by the Army Research Office through Cooperative Agreement Number W911NF-20-2-0214. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

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

[2] Ofra Amir, Finale Doshi-Velez, and David Sarne. 2019. Summarizing agent strategies. *Autonomous Agents and Multi-Agent Systems* 33, 5 (2019), 628–644.

[3] Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. 2019. Explainable agents and robots: Results from a systematic literature review. In *18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*. International Foundation for Autonomous Agents and Multiagent Systems, 1078–1088.

[4] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 5 (2009), 469–483.

[5] Ross D. Arnold, Hiroyuki Yamaguchi, and Toshiyuki Tanaka. 2018. Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *Journal of International Humanitarian Action* 3, 1 (12 2018), 18.

[6] Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. 2011. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, Vol. 33.

[7] S. Baron-Cohen. 1991. Precursors to a theory of mind: Understanding attention in others. *Whiten, Andrew (ed.), Natural theories of mind* (1991), 233–251.

[8] Paul Baxter, Grzegorz Cielniak, Marc Hanheide, and Pål From. 2018. Safe Human-Robot Interaction in Agriculture. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* (Chicago, IL, USA) *(HRI '18)*. Association for Computing Machinery, New York, NY, USA, 59–60. https://doi.org/10.1145/3173386.3177072

[9] G. S. Becker. 1976. *The economic approach to human behavior*. University of Chicago press.

[10] Or Biran and Courtenay V. Cotton. 2017. Explanation and Justification in Machine Learning : A Survey Or.

[11] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games* 4, 1 (2012), 1–43.

[12] Ruth Byrne and P.N. Johnson-Laird. 2009. 'If' and the problems of conditional reasoning. *Trends in cognitive sciences* 13 (07 2009), 282–7. https://doi.org/10.1016/j.tics.2009.04.003

[13] Anthony R Cassandra, Leslie Pack Kaelbling, and Michael L Littman. 1994. Acting optimally in partially observable stochastic domains. In *Aaai*, Vol. 94. 1023–1028.

[14] Tathagata Chakraborti, Sarath Sreedharan, Sachin Grover, and Subbarao Kambhampati. 2019. Plan Explanations as Model Reconciliation–An Empirical Study. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 258–266.

[15] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. 2020. The Emerging Landscape of Explainable Automated Planning & Decision Making.. In *IJCAI*. 4803–4811.

[16] Thomas Griffiths, Nick Chater, Charles Kemp, Amy Perfors, and Joshua Tenenbaum. 2010. Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in cognitive sciences* 14 (08 2010), 357–64.

[17] Ahmed Hassanein, Mohanad Elhawary, Nour Jaber, and Mohammed El-Abd. 2015. An autonomous firefighting robot. In *2015 International Conference on Advanced Robotics (ICAR)*. 530–535. https://doi.org/10.1109/ICAR.2015.7251507

[18] Bradley Hayes and Julie A Shah. 2017. Improving Robot Controller Transparency Through Autonomous Policy Explanation. In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*. ieeexplore.ieee.org, 303–312.

[19] Thomas Hellström and Suna Bensch. 2018. Understandable Robots. *Paladyn, Journal of Behavioral Robotics* 9 (07 2018), 110–123. https://doi.org/10.1515/pjbr-2018-0009

[20] Ayanna Howard and Jason Borenstein. 2018. The ugly truth about ourselves and our robot creations: the problem of bias and social inequity. *Science and engineering ethics* 24, 5 (2018), 1521–1536.

[21] Sandy H. Huang, David Held, Pieter Abbeel, and Anca D. Dragan. 2019. Enabling Robots to Communicate their Objectives. *Autonomous Robots* 43, 2 (February 2019).

[22] Tobias Huber, Katharina Weitz, Elisabeth André, and Ofra Amir. 2021. Local and global explanations of agent behavior: Integrating strategy summaries with saliency maps. *Artificial Intelligence* 301 (2021), 103571.

[23] Philip Johnson-Laird. 2010. Mental models and human reasoning. *Proceedings of the National Academy of Sciences of the United States of America* 107 (10 2010), 18243–50. https://doi.org/10.1073/pnas.1012933107

[24] Lynn Julien-Schultz, Nancy Maynes, and Cilla Dunn. 2010. Managing Direct and Indirect Instruction: A Visual Model to Support Lesson Planning in Pre-Service Programs. *The International Journal of Learning: Annual Review* 17 (2010), 125–140.

[25] Omar Khan, Pascal Poupart, and James Black. 2009. Minimal sufficient explanations for factored markov decision processes. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 19.

[26] Joseph Kim, Christian Muise, Ankit Shah, Shubham Agarwal, and Julie Shah. 2019. Bayesian Inference of Linear Temporal Logic Specifications for Contrastive Explanations.. In *IJCAI*. 5591–5598.

[27] Sau lai Lee, Ivy Yee man Lau, S. Kiesler, and Chi-Yue Chiu. 2005. Human Mental Models of Humanoid Robots. In *2005 IEEE International Conference on Robotics and Automation*. 2762–2772.

[28] Michael S. Lee, Henny Admoni, and Reid Simmons. 2021. Machine Teaching for Human Inverse Reinforcement Learning. *Frontiers in Robotics and AI* 8 (2021), 188. https://doi.org/10.3389/frobt.2021.693050

[29] Tim Miller, Piers D. L. Howe, and Liz Sonenberg. 2017. Explainable AI: Beware of Inmates Running the Asylum Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences. *ArXiv* abs/1712.00547 (2017).

[30] Kevin A. Nguyen, Jenefer Husman, M. A. Trujillo Borrego, Prateek Shekhar, Michael J. Prince, Matt DeMonbrun, Cynthia J. Finelli, Charles Henderson, and Cindy K. Waters. 2017. Students' Expectations, Types of Instruction, and Instructor Strategies Predicting Student Response to Active Learning. *International Journal of Engineering Education* 33 (2017), 2–18.

[31] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. 2018. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics* 7, 1-2 (2018), 1–179.

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

[33] Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

[34] Tiia Ruutmann and Hants Kipper. 2011. Teaching Strategies for Direct and Indirect Instruction in Teaching Engineering. In *2011 14th International Conference on Interactive Collaborative Learning*. 107–114.

[35] David Silver. [n.d.]. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. ([n. d.]).

[36] David Silver, Julian Schrittwieser, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the game of Go without human knowledge. *Nature* 550 (10 2017), 354–359.

[37] Aaron Steinfeld, Odest Chadwicke Jenkins, and Brian Scassellati. 2009. The oz of wizard: simulating the human for interaction research. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*. 101–108.

[38] Roykrong Sukkerd, Reid Simmons, and David Garlan. 2020. Tradeoff-Focused Contrastive Explanation for MDP Planning. *arXiv preprint arXiv:2004.12960* (2020).

[39] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[40] Aaquib Tabrez, Shivendra Agrawal, and Bradley Hayes. 2019. Explanation-based Reward Coaching to Improve Human Performance via Reinforcement Learning. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 249 – 257.

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

[42] Giulia Vilone and Luca Longo. 2020. Explainable Artificial Intelligence: a Systematic Review. (05 2020).

[43] Lihui Wang, Sichao Liu, Hongyi Liu, and Xi Vincent Wang. 2020. Overview of Human-Robot Collaboration in Manufacturing. In *Proceedings of 5th International Conference on the Industry 4.0 Model for Advanced Manufacturing*, Lihui Wang, Vidosav D. Majstorovic, Dimitris Mourtzis, Emanuele Carpanzano, Govanni Moroni, and Luigi Maria Galantucci (Eds.). Springer International Publishing, Cham, 15–58.

[44] Yusen Zhan, Anestis Fachantidis, Ioannis Vlahavas, and Matthew E. Taylor. 2014. Agents Teaching Humans in Reinforcement Learning Tasks. In *International Conference on Autonomous Agents and Multiagent Systems*.