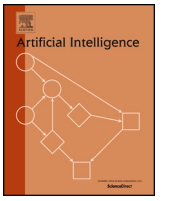
Artificial Intelligence 330 (2024) 104107

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



"Guess what I'm doing": Extending legibility to sequential decision tasks

Miguel Faria [a,b,*], Francisco S. Melo [a,b], Ana Paiva [a,b]

[a] Tecnico Lisboa, Lisbon, Portugal
[b] INESC-ID, Lisbon, Portugal

ARTICLE INFO

Keywords:
Legible decision making
Planning
Legibility
Agent communication

ABSTRACT

In this paper we investigate the notion of *legibility* in sequential decision tasks under uncertainty. Previous works that extend legibility to scenarios beyond robot motion either focus on deterministic settings or are computationally too expensive. Our proposed approach, dubbed PoLMDP, is able to handle uncertainty while remaining computationally tractable. We establish the advantages of our approach against state-of-the-art approaches in several scenarios of varying complexity. We also showcase the use of our legible policies as demonstrations in machine teaching scenarios, establishing their superiority in teaching new behaviours against the commonly used demonstrations based on the optimal policy. Finally, we assess the legibility of our computed policies through a user study, where people are asked to infer the goal of a mobile robot following a legible policy by observing its actions.

1. Introduction

Interaction between humans and agents/robots can greatly benefit from the ability to reason about each others' intentions—inferring what the other is trying to do and what its objectives are. During interactions, between an agent and a human, clarity of intentions allows for humans more time to react and adapt to the agent's actions and intentions. Thus, by being clearer about intentions and objectives, interactions between humans and agents can become safer both [2,13]. Take, for example, a recommendation system—like Amazon—suggesting products of interest to a client. By increasing the agent's clarity, it becomes easier for the user to understand what searches gave origin to the recommendation; and thus, may become more trusting of the system. Another example of the usefulness of improving the clarity of intentions behind the actions of an agent is in human-robot interaction. Consider a warehouse human supervisor, managing a fleet of robots moving objects around the warehouse; by improving the robots' clarity, the supervisor better understands where each robot is going and can issue more correct commands to each robot. Besides efficiency and safety, clarity of intentions is also useful for machine teaching applications [57]. Machine teaching, as defined by Zhu et al. [58], is a problem where a teacher must teach a target model θ^* to a learner using examples, choosing the best examples to show the learner. Machine teaching has many interesting applications, such as helping human teachers to pick the best examples to show in class or to train cybersecurity applications to identify attack patterns and learn the best defense policies [20,58]. Thus, clarity and transparency are two important characteristics in machine teaching, so the learner can learn the model requiring fewer examples. Ho et al. [29] explored how the clarity of the demonstrations impacts a learner's learning experience; and concluded that humans prefer demonstrations that better disambiguate the goal, instead of those that simply optimally teach how to reach the goal.

* Corresponding author.
E-mail addresses: miguel.faria@tecnico.ulisboa.pt (M. Faria), fmelo@inesc-id.pt (F.S. Melo), ana.paiva@inesc-id.pt (A. Paiva).

https://doi.org/10.1016/j.artint.2024.104107
Received 20 September 2022; Received in revised form 22 February 2024; Accepted 25 February 2024

Available online 7 March 2024

In the human-robot interaction (HRI) literature, several works have explored the communication of intentions using speech [47, 50], gaze [8,15], and movement [19,23]. In this work, we address the problem of conveying intention through *action*, which is closely related to the aforementioned works that explore the communication of intention through movement. In particular, we are interested in the notion of *legibility*, introduced by Dragan, Lee, and Srinivasa [18], that measures to what extent a user is able to infer the goal of a robot by observing a snippet of the robot's movement.

A legible movement is characterized not by its *efficiency* in reaching the goal, but by its *distinctiveness*, i.e., by how much it is able to disambiguate the actual goal of the movement from other potential goals. In the original work of Dragan, Lee, and Srinivasa [18], legibility is expressed by the probability of the goal given the movement, i.e.,

$$L(\text{movement}) = P(\text{Goal} \mid \text{Movement snippet}).$$

Legibility has been widely explored in human-robot interaction to improve a robot's expressiveness through movement [19,34]. Legibility differs from the traditional problem of goal recognition as in [24]. In the problem of goal recognition, the agent tries to infer the goal of another agent through a sequence of observations [4]; while, in the problem explored in this work, the actor agent tries to infer the best actions that allow an observer agent to infer the actor's goal. So, in this work we are exploring the counterpart to the problem of goal recognition.

Transparency of machine learning applications and artificial agents' decision-making processes has been an important topic of research, to create autonomous systems that can be better understood by the humans that interact with them. Strouse et al. [53] explored how to use information sharing and hiding to improve agent efficiency in multi-agent scenarios, both in cooperative and competitive tasks. Chakraborti, Sreedharan, and Kambhampati [11] propose a planning technique named MEGA (Multimodel Explanation Generation Algorithm), capable of balancing generated plans with the required explanations to make a plan optimal for an external observer, with a possibly different world model. So, MEGA makes the generated plans optimal both for the agent and external observers that might not share the same model as the agent.

The focus on improving the transparency and explainability of autonomous systems has been one of the main drives for the application of legibility beyond robotic motion [5]. Habibian and Losey [28] developed a framework for the legible allocation of subtasks in teams composed of robots and humans. The proposed framework leverages the notion of legibility to identify subtask allocations, in interactions without predetermined roles, so that a human can understand quicker from the robots' behaviours. With the proposed framework the humans interacting with the robotic team are shown to be more prone to keep collaborating with the robot team. MacNally et al. [38] developed a framework for legible decision-making in deterministic scenarios. The proposed framework generates plans that focus on increasing simultaneously the plan's utility and transparency. To that end, the framework selects the sequence of actions that best induces the observations required so that an observer's belief more quickly converges to the intended goal belief. Both approaches showcase the benefits of legible behaviours when the outcome of the actions of the agent is deterministic, much as in the motion planning setting of Dragan and Srinivasa [17].

Miura, Cohen, and Zilberstein [43] further extended the notion of legibility to scenarios of planning under uncertainty, introducing *legible Markov decision process*. In legible MDPs, the planning agent reasons about the observer's belief regarding the goal of the observed actions, using the multiagent framework of *interactive POMDPs* (I-POMDPs) [27]. Unfortunately, the planning complexity of Miura's legible MDPs is similar to that of partially observable Markov decision process (POMDPs) [36,39], making it impractical for large-scale problems.

In this work, we propose an alternative formulation of legibility in sequential decision-making problems under uncertainty. Our framework, dubbed *Policy legible MDP* (PoLMDP), avoids explicit theory of mind and, instead, defines an alternative MDP reward function that is akin to the legibility score in the original work of Dragan and Srinivasa [17]. Using the new reward function, our agent can compute a legible policy by solving a standard MDP, which provides a tractable alternative to Miura's legible MDPs. We show that PoLMDPs generate legible behaviours significantly faster than Miura's legible MDPs while attaining similar levels of legibility according to either notion of legibility. We also validate the legibility of the policies computed from PoLMDPs in a user study, where human users are asked to identify the goal of an agent's actions computed from a PoLMDP. Finally, and motivated by the findings of Ho et al. [29], we explore the impact that legible policies—computed with PoLMDP—can have when teaching other agents. Specifically, we use PoLMDP policies as demonstrations for an inverse reinforcement learning agent [44,49] as the learner agent and show that PoLMDP demonstrations lead to faster learning when compared with the commonly used optimal policies. Such result leads to the conclusion that PoLMDP policies are better at conveying task information than optimal policies (i.e., they are, indeed, more legible). We use a MazeWorld like scenario in our experiments because this type of scenario is widely used in related literature; as well as serving as the foundation for other more complex environment, thus establishing a baseline for more complex settings.

2. Background

This section introduces key concepts and notation used in the remainder of our work.

2.1. Markov decision processes

A *Markov decision process (MDP)* is a model for sequential decision problems in stochastic environments. An MDP M is defined as a tuple $\langle \mathcal{X}, \mathcal{A}, P, r, \gamma \rangle$, with $\mathcal{X}$ the state space; $\mathcal{A}$ the action space; P the state transition probabilities, where $P(y \mid x, a)$ indicates the probability of moving from state x to state y upon executing action a; $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is the reward function, where $r(x, a) =$

$\mathbb{E}\left[R_t \mid X_t = x, A_t = a\right]$, and X_t, A_t and R_t respectively denote the state, action and reward at time t; $\gamma \in [0,1)$ is a discount factor, indicating the relative importance of future rewards against present rewards.

Solving an MDP amounts to computing an *optimal policy* π^*. A policy is a mapping from states to actions describing which action the agent should take in each state, and we can define the *value* associated with a policy as

$$v^{\pi}(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \mid X_t = x\right],$$

where X_t and R_t are the state and reward at time step t, respectively. The optimal policy is such that $v^{\pi^*}(x) \geq v^{\pi}(x)$ for all $x \in X$ and all policies π. The value associated with the optimal policy is denoted v^*, and we define the *optimal Q-function* as

$$q^*(x,a) = r(x,a) + \gamma \sum_{y \in X} P(y \mid x,a) v^*(y).$$

The optimal Q-function can be computed in polynomial time using dynamic programming [46], and the optimal policy can be computed from q^* simply as $\pi^*(x) = \operatorname{argmax}_{a \in A} q^*(x,a)$.

In the rest of this work, we refer to a *goal* as the reward function r of an MDP. So, by defining a different reward function, we can define a different goal for the MDP.

2.2. Inverse reinforcement learning

Inverse reinforcement learning is the problem of recovering/learning a reward for a rewardless MDP, $\langle \mathcal{X}, \mathcal{A}, P, \gamma \rangle$, given the corresponding optimal policy [44]. In gradient IRL (GIRL) [37], the learner receives a demonstration consisting of N independent state-action pairs, $D = \left\{(x_n, a_n), n = 1, \ldots, N\right\}$, and computes a reward r^* so as to maximize the likelihood of D, i.e.,

$$r^* = \operatorname*{argmax}_r \prod_{n=1}^{N} P(x_n, a_n \mid r),$$

where

$$P(x_n, a_n \mid r) = \frac{1}{Z} \exp(\eta q_r^*(x_n, a_n)),$$

and q_r^* is the optimal Q-function given the reward r, Z is a normalization constant and η is a tunable parameter. GIRL computes the reward r^* using standard gradient ascent.

2.3. Legibility

Legibility is the property that describes how "readable" a movement's objective is and is inspired by the principle of rational action [48], which states that a "*rational agent will act efficiently and justifiably to achieve its goals.*" The legibility of a movement is quantified as the probability that a human observer will associate a specific objective, g, to the robot's observed snippet of movement, $\xi_{x_0 \to x_t}$, with x_0 the robot's starting pose and x_t the robot's pose at time t. The goal inference can be defined as

$$\mathcal{I}_L(\xi_{x_0 \to x_t}) = \operatorname*{argmax}_{g \in G} P(g \mid \xi_{x_0 \to x_t}), \tag{1}$$

where G is the set of possible goals of the robot's movement. Using Bayes' Rule

$$P(g \mid \xi_{x_0 \to x_t}) \propto P(\xi_{x_0 \to x_t} \mid g) P(g), \tag{2}$$

with $P(g)$ the prior on the goals and $P(\xi_{x_0 \to x_t} \mid g)$ the probability of the observed trajectory snippet when the robot is moving towards goal g. In the original work of Dragan, Lee, and Srinivasa [18], the latter is expressed as a maximum entropy distribution of the form

$$P(\xi_{x_0 \to x_t} \mid g) = \frac{\exp(-C(\xi_{x_0 \to x_t}) - C(\xi^*_{x_t \to g}))}{\exp(-C(\xi_{x_0 \to g}))}, \tag{3}$$

where $\xi^*_{x_t \to g}$ denotes the optimal trajectory from the robot's pose at time t to the goal pose g and $C(\xi)$ is the cost associated with trajectory ξ. (3) captures how a human associates a goal g to the observed trajectory snippet $\xi_{x_0 \to x_t}$ as being higher the closer the robot's moves to the goal g and farther from other possible goals.

3. Related work

The work we present is heavily based on the notion of legibility. As such, we proceed to present the most relevant works that show the effects of applying this notion to intelligent agents. We start by exploring the applications and effects of legibility in robotics since it was the area where the notion was first applied. Then, we move to works that have brought the use of legibility beyond robotics and to more general agent applications.

3.1. Legibility in robotics

The increasingly ubiquitous presence of robots and autonomous agents in society has made paramount research on how to make these artificial entities clearer and transparent regarding their intentions [3,12,26]. One of the main trends of research is on improving agent transparency through implicit communication [31,35,52], namely using robot motion as a means of conveying intentions [22,30,34,54]. Legible motions were proposed by Dragan, Lee, and Srinivasa [18] as a type of expressive motion, and take advantage of the principle of rationality [48] combined with principles from animation to shape robotic motion in a way that helps to disambiguate a robot's objectives. Since the introduction of legible motions, they have been shown capable of improving a robot's expressiveness [19,21].

Dragan et al. [19] explored the impacts of legibility in one-on-one interactions between a human and a robot, when compared to motions that improve the robot's efficiency. In this work, the robot-human team had to fulfill orders in a coffee-shop-like scenario and the comparison between the different types of movements showed that humans found it easier and more efficient to interact with legible motions instead of other motions focused on efficiency metrics like covered distance or expended energy.

While early works on legibility [18,19] have explored legibility as a property of a specific task, captured and optimized via a specific cost function, adapting such approach to different tasks and different classes of users is non-trivial and requires new cost functions to be designed. In an attempt to tackle this problem, Busch et al. [9] designed an approach where a robot learns to adapt its behaviour to become more legible through repeated interactions. The designed approach uses reinforcement learning (RL) to learn a behaviour model for how to interact legibly with a human. To create this model the robot needs a training phase, where it interacts with a human and learns how to perform the task and be more legible while performing it. With the developed approach, the authors showed that a robot can learn how to adapt its movements to become more legible and how to generalize this legible behaviour to different tasks without having to create a new model of interaction.

Faria et al. [21,23] considered the impact of legibility in multi-party Human-Robot Interaction (HRI) scenarios. Faria et al. [21] present a user study with a robot serving cups of water to groups of three human partners, who do not know the order in which the robot is going to serve them. The results of the study show that using only efficient movements led to worse collaboration between the humans and the robot than when the robot uses some kind of legible movements. When the robot focuses only on using efficient movements, the humans would sometimes even get confused regarding whom the robot was serving and would get in the way of one another. Despite these initial positive results, there were some situations where the legible motions also caused some confusion on the users, because the participants' perspective led them to believe that the movement was towards another person. So in a posterior work [23], the authors expanded the ideas from Nikolaidis, Dragan, and Srinivasa [45] to build an approach capable of generating legible motions in multi-party scenarios that took the different perspectives of the users into consideration. The obtained motions improved the legibility for all the participants, independently of their perspectives on the movement. Additionally, a user study conducted over M-Turk, showed that this new approach was capable of generating legible motions that maintained their legibility when observed from different perspectives, leading to humans better predicting the goal of the robot's motions.

Legible motions have also been used in mobile navigation and to improve robot motions for object manipulation. Mavrogiannis, Thomason, and Knepper [40] researched the use of legible motions in the planning of socially aware robotic navigation. In their work, the authors developed a framework—*Social Momentum*—that combines an efficiency metric, to drive the robot towards an intended goal, with legible motions, when the robot needs to avoid another agent in the same workspace. Using the social momentum framework, the authors showed that a robot is capable of navigating in crowded workspaces and avoiding collisions. This performance was achieved by adopting legible behaviours when passing by another agent, which allowed other agents to understand which direction the robot intended to move. Wallkötter, Chetouani, and Castellano [55] proposed a framework named "SLOT-V", a supervised learning approach to learn an observer's model and create motions that better convey the robot's intentions during a manipulation task. Using SLOT-V, the authors showed that a robot can generalize learned observer models to different trajectories in the same environment, but also for new environments including unseen environments. In other words, SLOT-V allows a robot can generate legible trajectories both for scenarios where it was trained and for scenarios it had little interaction with.

3.2. Transparency of AI systems

Most approaches to improve AI systems' transparency have focused on making more transparent technical aspects of AI systems, such as the reason behind application failures [6,10,16] or the inference behind complex decision processes like deep learning approaches [1,51,56]. However, most such approaches focus on the interpretability of AI systems and on making agents explainable from a user perspective. In other words, the agent justifies its actions by presenting its reasoning after deciding, instead of making the reasoning clear while deciding. However, the growing intertwining of artificial intelligence (AI) with society led these systems to become more than simple tools and applications we use; instead, they become peers and collaborators with whom humans interact. In this sense, AI systems need to be transparent during the entire decision process instead of just at the end of the interaction [5]. *Ad hoc* teamwork problems [25,41] showcase the importance of transparency during the entire task or decision process instead of just one prompted. In ad hoc teamwork, a team of agents has to learn to cooperate on the fly without any prior coordination or joint training of the team members. Since ad hoc teamwork requires agents to interact without any prior coordination protocols established or joint training, the need for clear and transparent behaviours is paramount so the different agents understand each others' goals and behave accordingly.

The success of legibility to shape motions to become more expressive has made the notion of legibility a good candidate to create more transparent AI systems [5]. Alonso and De La Puente [3] present a mini-review of literature that gathers several works on

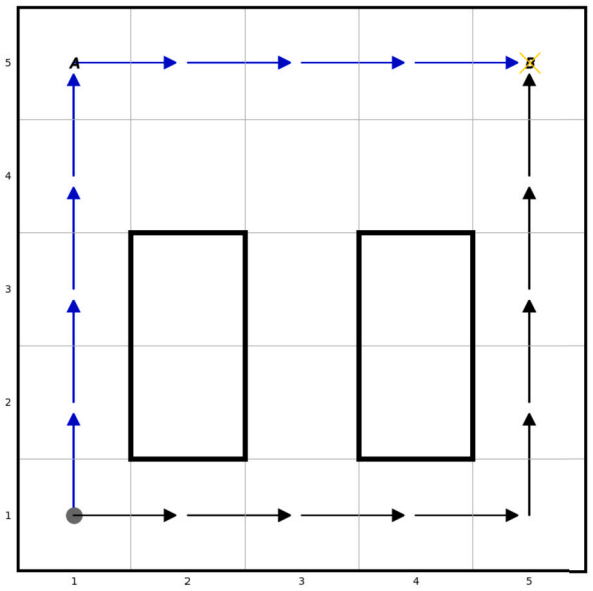


Fig. 1. Example of maze-like environment with two goals, A and B. The blue arrows indicate a possible action sequence following an optimal policy, while the black arrows indicate an action sequence following a legible policy (which is also optimal). (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

transparency in shared autonomy workspace and explores different methods of increasing transparency. In this review, the authors conclude that legibility is one of the most common methods to increase transparency when modelling the system's behaviours.

The usefulness of legibility in making more transparent robots has also led to legibility being used beyond robotic motions. Habibian and Losey [28] designed a framework to divide the subtasks of a more overarching task between humans and robots, using legible and fair allocations. The designed framework performs a bilevel optimization: the first level is an optimization for the allocation of the subtasks, such that the human clearly understands what the robots' are doing and what subtasks are left for the human to do; then the second level of optimization focuses on optimizing the robots' motions to create motions that are in line with the decided allocation and thus, when observed by the human, allow him to understand what are the robots' intentions. With this work, we can observe how the notion of legibility can be applied beyond robotic motions to create allocations that are easily understood by humans.

The application of legibility to decision-making has also been explored by MacNally et al. in [38]. In this work, the authors formalize the decision-making problem as a Goal POMDP [32], where the agent's goal is to choose the actions that transform an initial belief b_0 into the goal belief b_G. Using this formalization, the authors design a method for action selection, in deterministic scenarios, that chooses the sequence of actions that constitutes the plan that achieves b_G and best disambiguates the intended goal state from other possible goal states.

The notion of legibility has also been applied to scenarios of planning under uncertainty. Miura, Cohen, and Zilberstein [43] present a formulation of legibility for MDPs, named *legible MDP*. In legible MDPs, the agent focuses on choosing, at each time step, the most optimal action that also maximizes the information conveyed to an observer about the agent's goal. To accomplish such optimal and legible behaviour, the agent reasons about the observer's belief regarding the objective of the history of observed agent actions. The observer's belief regarding the agent's intentions is modelled using the multiagent framework of *interactive POMDPs (I-POMDPs)* [27]. By reasoning about the observer's—using this reasoning to drive the planning algorithm—the agent can derive a *legible policy* that disambiguates the agent's goal [42]. The legible policy is obtained by iteratively updating the estimated observer's belief about the agent's goal given the current history of actions. Then, with the updated belief simulate the possible actions to find the one that best disambiguates the agent's goal (for example, using UCT [33]). The results of a user study conducted by the authors show that the resulting legible policies are capable of better transmitting the agent's intentions than using standard optimal policies. However, the nature of legible MDPs makes their planning complexity similar to that of POMDPs, limiting its applications to small-scale state spaces as the planning can become intractable in large-scale state spaces.

4. Policy legible Markov decision problem

An optimal policy π^* describes the most efficient way an agent can solve an MDP. However, the optimal policy does not guarantee the decisions to be clear to an observer and may cause doubts regarding what the agent is trying to accomplish. In Fig. 1 we can observe, in blue, a possible sequence of actions prescribed by an optimal policy, for an agent moving in a maze world scenario, towards the objective B. We can observe that the optimal policy leads the agent to objective B while going through A. An observer that did not know the robot's intentions could, upon observing the initial actions of the agent, confuse the robot's goal to be A rather than B.

We propose to adapt the notion of legibility to MDPs to yield policies that offer both a solution with a high expected reward and make clear the agent's current objectives. To achieve such policies we introduce the *policy legible Markov decision process (PoLMDP)*. A PoLMDP is defined in the context of an environment with N different objectives, each represented by a different reward function $r_n, n = 1, \ldots, N$, and thus defining a different MDP—$MDP_1, MDP_2, \ldots, MDP_N$. Each reward function, r_n, is defined as in Section 2.1 for the rest of the article. We consider that $r_n(x, a) \in [0, 1]$, for $x \in \mathcal{X}$ and $a \in \mathcal{A}$, with $r_n = 1$ at the goal state. A PoLMDP is then described as a tuple $\langle \mathcal{X}, \mathcal{A}, P, r_{\text{leg}}, \gamma, \beta \rangle$, with $\mathcal{X}$, $\mathcal{A}$, P and γ are as in Section 2.1, r_{leg} defines the legible reward function and measures each action's legibility in each state, and β is a non-negative constant that defines how close the legible function follows the optimal expected reward.

Following the original definition of Dragan and Srinivasa [17], we define r_{leg} to measure the legibility of an action a in state x by evaluating how likely it is that a is executed in state x given that the current goal is defined by reward r_n, in opposition to performing the same action when trying to achieve another possible objective. In Fig. 1 we can observe, in black, the decision sequence prescribed by a PoLMDP policy, for an agent moving in a maze-like scenario towards objective B. The action sequence prescribed by PoLMDP focuses on moving the agent away from objective A as soon as possible while getting closer to B.

We define our legible reward function r_{leg} for a target reward r_n as

$$r_{\text{leg}}(x, a) = P(r_n \mid (x, a)), \tag{4}$$

where

$$P(r_n \mid (x, a)) \propto P(a \mid x, r_n) P(r_n).$$

This is similar to the definition of Dragan and Srinivasa [17] in (2). However, instead of observing a movement snippet, we observe the current state and action performed as indicators of the robot's intentions. We choose this representation because, in a MDP, each decision step is independent of the past given the current state, and as such the probability of executing an action a in state x is not influenced by possible states and actions that preceded the current state. This way, we need only observe the current state-action pair to infer the robot's intentions. Considering a uniform distribution as the prior on the probability of observing each goal, we can simplify the previous expression to

$$r_{\text{leg}}(x, a) = P(a \mid x, r_n). \tag{5}$$

This probability reflects how probable an action a is in a state x when trying to achieve one specific objective against when it tries to achieve another possible objective. We follow the reasoning of Dragan and Srinivasa [17], combining the notion of action understanding as inverse planning proposed by Baker, Saxe, and Tenenbaum [7] with the maximum-entropy principle proposed by Ziebart et al. [59] to define the likelihood of an action a occurring in a state x for a specific goal. So, we define $P(a \mid x, r_n)$ as

$$P(a \mid x, r_n) = \frac{\exp(\beta q_n^*(x, a))}{\sum_{m=1}^{N} \exp(\beta q_m^*(x, a))}, \tag{6}$$

with β the parameter in the PoLMDP description tuple and q_n^* the optimal Q-function for MDP n. From (6), an action a is more legible the larger the gain of executing it in MDP n in comparison to the gain of performing a for other possible MDPs. The policy obtained using the reward in (5) promotes actions that guide the agent towards its intended goal while increasing the agent's expressiveness.

4.1. Relation with other approaches

The approach taken in our PoLMDP is closest to the legible MDP of Miura, Cohen, and Zilberstein [43]. Both approaches offer formulations of legibility for MDPs, differing in the approach taken to compute the legible policy. In the formulation of Miura et al., the legible MDP builds a reward function from a belief b_t that, at each time step t, translates the observer's inferred goal from the actions of the agent observed up to time step t. The authors propose to update the belief b_t as follows

$$\begin{aligned} b_t(r_n) &= P(r_n | x_{t-1}, a_{t-1}, x_t, b_t) \\ &= \frac{\hat{P}_n\left(x_t \mid x_{t-1}, a_{t-1}\right) \hat{\pi}_n\left(a_{t-1} \mid x_{t-1}\right) b_{t-1}(\theta)}{\sum_{m=1}^{N} \hat{P}_m\left(x_t \mid x_{t-1}, a_{t-1}\right) \hat{\pi}_m\left(a_{t-1} \mid x_{t-1}\right) b_{t-1}\left(r_m\right)}, \end{aligned}$$

with h_t the agent's history up to time step t, $\hat{P}_n$ is the assumed transition for MDP n and $\hat{\pi}_n$ the agent's policy for MDP n. The reward is computed from either the Euclidean distance or KL-divergence between the observer's (estimated) belief, b_t, and the belief b^* translating the correct goal, *i.e.*,

$$dist(b_t, b^*) = D_{KL}(b_t \,||\, b^*) = \sum_{n=1}^{N} b_t(r_n) \log \frac{b_t(r_n)}{b^*(r_n)}. \tag{7}$$

The dependence of the reward on b_t introduces a dependence on the history of the process, rendering legible MDPs not amenable to the use of standard MDP or POMDP solution techniques. As discussed by Miura and Zilberstein [42], the legible MDP can be derived as a special case of an I-POMDP, an extension of POMDP to multi-agent settings. Thus, the complexity of solving a legible MDP is similar to the complexity of solving a POMDP (PSPACE-complete [14]).

In fully observable Markovian settings, an observer can either decide on the next action using the full history of interaction—all state-actions pairs since the environment started—or, since the setting verifies the Markov property, it requires only the most recent observation to decide on the next action. Our approach to legible decision-making assumes that, in Markov settings, the observer's belief regarding the agent's objectives is independent from previous states and actions. So, PoLMDPs consider that the observer's belief regarding the acting agent's goal is independent of the history and influenced only by the last observed state-action pair of the agent. Thus making the legibility of an agent's decision directly dependent on the saliency of the state-action pair

The reward function of PoLMDP circumvents the dependence on the history using (6) as its reward function. Furthermore, (6) shares similarities with the reward function of Miura et al. PoLMDPs consider an implicit distance between beliefs given by the *Total Variation* (TV) distance

$$dist(b_t, b^*) = \frac{1}{2} D_{TV}(b_t, b^*) = \frac{1}{2} \sum_{\theta \in \Theta} |b_t(\theta) - b^*(\theta)|,$$

where b^* is an indicator function for the agent's goal. Miura et al., on the other hand, use the KL-divergence in (7). Both are special cases of f-divergence, where the KL-divergence uses $f(x) = x \log x$, and we use $f(x) = |x - 1|$. Avoiding dependence on the agent's history allows for PoLMDP to be solved using the same methods as traditional MDPs; thus the complexity of solving a PoLMDP is the same as that of solving an MDP (PSPACE) and much lower than that of L-MDP. However, not depending on the history of interaction has the limitation of the legible policy being more general and agnostic to the human the agent interacts with. Being agnostic to the humans in specific that the agent interacts with, allows for the framework to be more flexibly applied to different scenarios and settings; however, by not looking at the interaction history and how the human react reduces the ability of the agent to adapt to human behaviours that are outside the norm.

5. Experimental evaluation

In this section, we present the evaluation of our framework. Our evaluation aims at exploring three questions:

1. "How does our framework performs compared to the framework of legible MDPs proposed by Miura et al. [43]?" We want to understand how PoLMDP stands with respect to other types of legible sequential decision-making frameworks;
2. "Given an agent (learner) trying to infer a task from examples from a teacher, are the actions selected by the PoLMDP policy more informative examples, about the task, than actions selected by the optimal policy?";
3. "In an interaction with human users, is an agent using a legible policy generated by PoLMDP able to convey its intentions faster than using a standard optimal policy?"

To answer these three questions we divided the evaluation in three parts.

We start by explaining the scenario used in the three evaluation parts, detailing the motivation for using the chosen scenario and the main decisions behind the adaptations in the scenario. After explaining the scenario, we present the three evaluations performed.

5.1. Evaluation scenario

In designing the scenario to use in our evaluation, we had two main concerns in mind. The first concern was that the scenario used could be the same across the three evaluations, knowing that we had both comparisons between legible frameworks in simulation scenarios and comparisons between agent policies in studies involving human users. Thus, the scenario had to allow a full qualitative comparison between the performance of different frameworks for legible decision-making; while, at the same time, being sufficiently simple and familiar to allow the interaction between human users and autonomous agents. The second concern was that the environment's dynamics should not impact the legibility of the executed actions because, during an interaction between a human and an agent, the human only observes the outcome of the actions of the agent and not the action itself. So, in the case of a failed action, the failure must not influence the legibility of the action; otherwise, we add artifacts and errors to the assessment of legibility by the human that cannot be measured.

Having these two concerns in mind, we use a maze world scenario, where a robot has to navigate and reach one of the various coloured areas scattered around the maze. These areas serve as the different goals that the agent (the mobile robot) may be trying to reach. The robot has available five different actions: moving up, moving down, moving left, moving right, and no operation. When performing a move action, there is a 15% chance that the action may fail, leaving the robot's position unchanged.

The maze world scenario chosen is also interesting because it can serve as the base to model different real-world scenarios, *e.g.*, a search and rescue mission, or a warehouse with robots retrieving different items to deliver. It is a common type of scenario in decision-theoretic literature and sufficiently familiar to allow a human user to easily grasp the task and how to solve it. Secondly, the 15% chance of failing and staying in the same place, although simple, introduces stochasticity without impacting the legibility of the robot's actions in case of a failure. Other options would be for the robot to veer off course in a failure since the robot could only move in four directions—up, down, left, and right—such an option would take the appearance of a sudden slip either sideways or backward. In an environment with nothing that would cause such slippage, from an interaction point of view this would introduce artifacts that could impact the legibility in unforeseen ways and thus impact the results of the user study.

5.2. Comparison with similar frameworks

We compare our framework with Miura's legible MDP [43], which is the only framework that addresses the problem of legible decision-making in stochastic environments.

As discussed in Section 3.2, legible MDPs requires the iterative updating of the observer's belief of the agent's goal; for that, it uses the history of the interaction to determine the action that best disambiguates the agent's goal. Thus, in order to solve a legible MDP, we follow the approach of the original paper and use UCT [33] to compute the legible policy from sequences of simulated actions.

5.2.1. Setup

Our comparison uses a maze world scenario and explores how each framework (PoLMDPs and legible MDPs) performs when we scale:

- The number of possible goals, keeping constant the number of states;
- The number of states in the world, keeping the number of goals constant.

The measures used to compare the two frameworks were: the average time taken to find a sequence of decisions from the initial state to the given goal; the average legibility value for each sequence, using both the definition of legibility used by PoLMDPs and Miura's legible MDPs.

To compare the performance with varying numbers of goals, we tested both frameworks on a 25×25 maze world with a number of possible goals between 3 and 10, making up 7 different world configurations and test scenarios.

To assess the scalability with the number of states we designed multiple maze world configurations, all with 6 possible goals, except for the smallest maze whose size could only accommodate 3 goals. The smallest maze was taken from the legible MDPs paper [43] and has 40 states (5×8). The rest of the mazes had the following dimensions: 100 states (10×10), 625 states (25×25), 1600 states (40×40), 2500 states (50×50), 3600 states (60×60) and 5625 (75×75). The environments were designed so that, independently of the size of the grid, the maze forced all paths to areas that were common to multiple goals (such as open areas or hallways that could lead to multiple goals) and areas that were more probable when heading to one specific objective.

For each test, we sampled 250 pairs (initial state, goal) for each world configuration, with the only requirement that the initial state was different from the goal state.

Both frameworks required the optimal solutions for all goals in order to compute the legible rewards. So, when recording the average time to find a sequence of legible actions, we did not consider the time taken to compute the optimal policies.

After sampling the 250 pairs for each world configuration, we ran those pairs through each framework, limiting their execution time to a maximum of 2 hours for each test pair. If in those 2 hours, a framework could not give a solution, we would mark that test pair as a failure. We chose the 2-hour threshold based on the average time it took a simple optimal agent to solve the most difficult test scenario: on average, a standard MDP solution for all possible objectives was found in 30 minutes, so we decided to give 4 times that time to account for the difference in time complexity of the two approaches. Also, from an interaction perspective, the 2-hour limit ensures that a user does not have to wait indefinitely for a solution from the system.

5.2.2. Results

The first step after running the simulations is to clean the gathered data, so we can have a similar number of trials across the testing conditions—PoLMDP and Legible MDP—and across testing scenarios. The data cleaning step is important given the difference in failed tests between the two test conditions, as seen in Figs. 4 and 7. The figures clearly show that, while PoLMDPs successfully completed all the runs, the legible MDPs had a significant number of failed runs. It was this asymmetry that prompted the data cleaning process, to allow for a balanced comparison between the two approaches—namely, a comparison between results featuring a similar number of runs, and in which failed runs do not skew the computation of the mean performance.

The data cleaning thus consisted of considering, for each of the two approaches, only 100 out of all the runs. This ensured that all the Legible MDP runs considered were valid and allowed for a clearer comparison with PoLMDP. The selection of the 100 trials for each test scenario was done in two steps.

In the first step, we removed from consideration all the failed runs. Then, if more than 100 runs remained for any of the conditions, we used the execution time as the selection criteria. First, we ordered the runs in both approaches by the time to get a solution and then removed the runs with higher execution times until both approaches had the same number of runs. The second step was the selection of runs: for the test scenarios with only 100 runs we took all the runs for each approach; for the test scenarios with more than 100 runs, we randomly selected 100 runs for each approach.

Regarding the evaluation of the average time taken to compute a solution, and the performance of the resulting policies according to each of the two legibility metrics, Figs. 2, 3 and 4 show the results for the goal scalability and Figs. 5, 6 and 7 show the results for the state scalability experiment. Figs. 2 and 5 show the average legibility of the obtained solutions for each approach, according to the PoLMDP's legibility criterion. Similarly, Figs. 3 and 6 show the average legibility of the obtained solutions for each policy, using the Legible MDP's legibility criterion. Finally, Figs. 4 and 7 show the average time each approach needed to compute a solution.

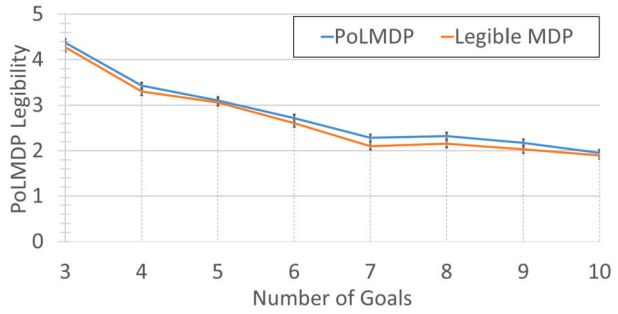


Fig. 2. Results for the PoLMDP legibility metric performance comparison between the PoLMDP framework against Miura's legible MDP, when we vary the number of possible goals in a mazeworld-like scenario.

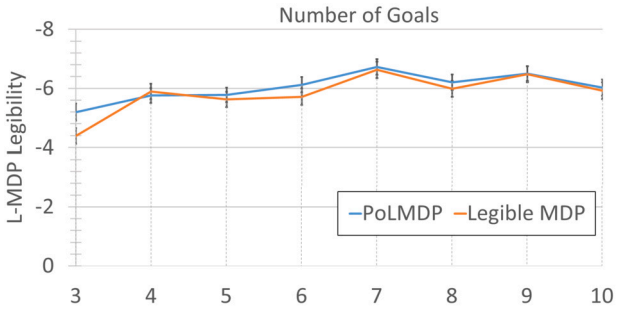


Fig. 3. Results for the Miura's legible MDP legibility metric performance comparison between the PoLMDP framework against Miura's legible MDP, when we vary the number of possible goals in a mazeworld-like scenario.

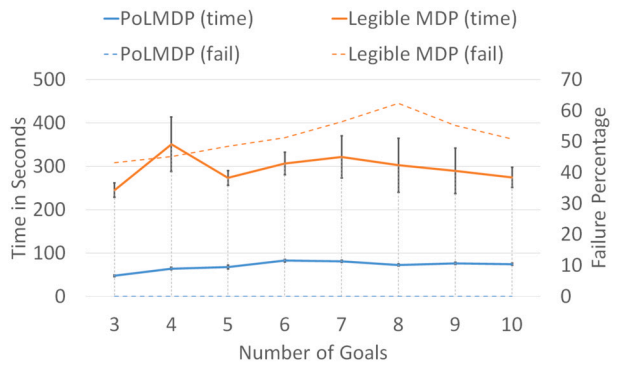


Fig. 4. Results for the time performance comparison between the PoLMDP framework against Miura's legible MDP, when we vary the number of possible goals in a mazeworld-like scenario. In continuous lines we show the average times, and, in dashed lines, the percentage of failed tests.

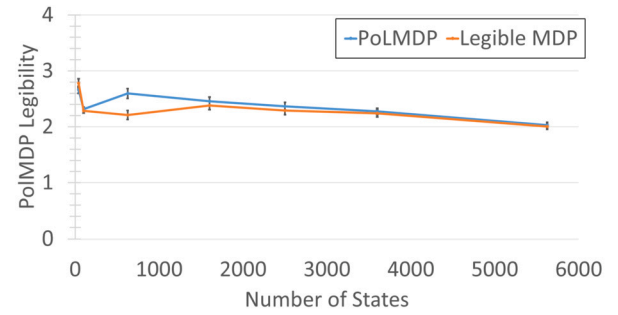


Fig. 5. Results for the PoLMDP legibility metric performance comparison between the PoLMDP framework against Miura's legible MDP, when we vary the number of states in the mazeworld scenario.

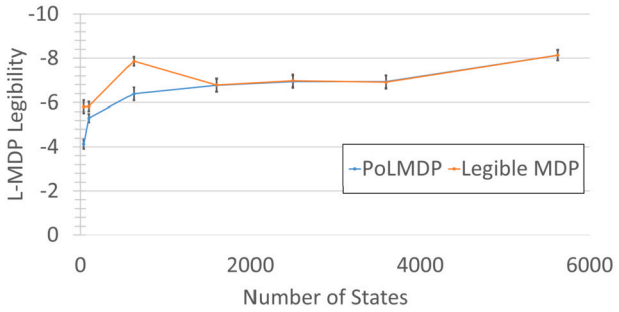


Fig. 6. Results for the Miura's legible MDP legibility metric performance comparison between the PoLMDP framework against Miura's legible MDP, when we vary the number states in the mazeworld scenario.

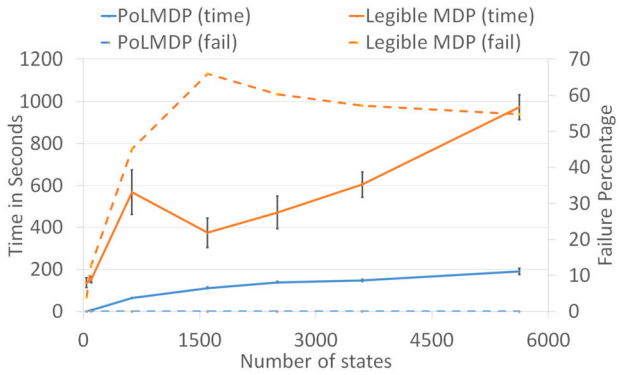


Fig. 7. Results for the time performance comparison between the PoLMDP framework against Miura's legible MDP, when we vary the number states in the mazeworld scenario. In continuous lines we show the average times, and, in dashed lines, the percentage of failed tests.

5.2.3. Discussion

The results of the first evaluation show interesting results. The first result that deserves analysis is the percentage of tests that failed. The legible MDPs had an average of 40-60% fail rate in most of the tests in both scalability tests. The only cases where such did not occur were on the tests with world configurations with small state spaces (≤ 100 states), where the failure rate was between 4% and 13%. On the other hand, looking at the results of PoLMDP, we can observe that the fail rate was always at 0%. PoLMDP was capable of always finding a solution in under 2 hours, for all scenarios considered. Such difference, although not surprising, sets the two methods widely apart, since Legible MDPs cannot effectively be used except for the smallest environments.

In robotic applications, we need our decision module to be able to give a solution quickly and the PoLMDP offers that capacity. These results are closely related to the results obtained for the average time to find a solution, where again PoLMDP outperformed legible MDP; the analysis of the average time also yields another interesting fact: with an increasing number of states, the PoLMDP's average time to find a solution increases at a much lower rate than that of Miura's legible MDP, which again hints at the PoLMDP's superior scalability.

Regarding the results of the legibility metrics, the two frameworks did not show a significant difference in the results obtained. Both frameworks performed better according to their respective metrics but without significant differences. This is an important result because it shows that the superior scalability of PoLMDP is not achieved at the cost of performance. Our results in this respect are particularly noteworthy since, looking at how the two frameworks assume the observer's model of the agent's intentions, we would expect that Legible MDP to achieve better average legibility scores since the agent's model adapts to how it expects the observer's inference to evolve during the interaction. However, with PoLMDP having a similar performance in legibility, we can conclude that integrating the history is not necessary to create equally legible behaviours and that the legibility of a behaviour is more closely tied to how unique the agent's action is to achieve the goal than to the sequence of actions of the agent. Such a conclusion is also in line with our discussion in Section 4, regarding the Markovian nature of the underlying decision process. Since the goal that the agent is trying to reach does not change with time, the information that the policy gives, regarding this goal, does not necessarily benefit from historical data to improve legibility.

In conclusion, the comparison between Miura's legible MDPs and PoLMDPs clearly shows the advantages of the latter over the former. Also, since the performance of both approaches in terms of legibility is essentially similar, in the remaining experiments we focus our analysis on a legible policy (PoLMDP) and an optimal one.

5.3. Ability to convey task information

In this section, we evaluate how informative the actions of the legible policy computed by PoLMDP are in the context of learning a task from examples. We consider an agent that observes the actions of a "teacher" and, from these actions, tries to infer the underlying task. In particular, we compare the ability of an agent to infer the target task when given actions selected by a PoLMDP policy as examples against the actions chosen by the optimal policy.

In this evaluation, we use a inverse reinforcement learning (IRL) agent as the learning agent to measure the efficiency of PoLMDP legible policies in conveying task information. The IRL agent uses examples given by an expert (our agent) to learn the expert's underlying reward function. Similarly to [37,49], we consider the case of a Bayesian IRL agent that maintains a set of possible rewards $\{r_1, \dots, r_M\}$ and, upon observing the set of examples $D = \{(x_m, a_m), m = 1, \dots, M\}$, computes $P(r_n \mid D), n = 1, \dots, N$. We then analyse how efficient our PoLMDP approach is, compared to a standard optimal policy, at selecting actions that allow the IRL agent to figure out the reward function being used.

5.3.1. Setup

In this evaluation, we ran two different tests. The first test evaluates how fast, in terms of the number of samples, a learning agent can figure out the task being taught when presented with a full sequence of state-action pairs leading to the solution of the task. Thus, in this first test, two learning agents observe a sequence of sampled state-action pairs $D = \{(x_m, a_m), m = 1, \dots, M\}$, where each state x_m is sampled from $P(\cdot \mid x_{m-1}, a_{m-1})$ and the corresponding action a_m is sampled from either a PoLMDP policy or from the optimal MDP policy. After each sample (x_m, a_m) is observed, the IRL agent tries to infer the task being taught—the one with maximum posterior probability given the observed examples.

The second test assesses the performance of the learner when the sequence of observed state-action pairs may not come from a single trajectory, *i.e.*, the sequence $D = \{(x_m, a_m), m = 1, \dots, M\}$ is built by sampling each state x_m independently at random from X, and the corresponding action a_m is again sampled from either a PoLMDP policy or from the optimal MDP policy. As in the first test, after each sample (x_m, a_m) is observed, the IRL agent tries to infer the task being taught. This second test represents those situations where the teaching examples may miss information, include noise, or where the expert is teaching how to solve specific situations within a task. We again had two learning agents, each observing samples from either the PoLMDP policy or the optimal policy. We sampled 20 examples for each approach, all corresponding to a single goal.

We used 4 different configurations of 10×10 mazes where we varied the goal positions and the configuration of the walls in the mazes. However, in all the configurations there were 6 possible goal locations that the robot could reach.

For the first test, we sampled each world configuration 250 initial positions. Then, for both testing conditions, we obtained ten sequences, with 20 steps each, between each initial condition and each one of the six possible objectives in the maze configuration. We then took each sequence and sequentially gave more examples of the same trajectory to the learning agent. Each time we gave a new example, we asked the learner to predict what was the robot's goal and registered if the prediction was correct or incorrect. For each initial position, we repeated this process for all 10 sequences for each goal and averaged the results per goal. To avoid having results influenced by previous runs, each time we changed the condition or started a new sequence, we re-instantiated the learning IRL agent. We chose sequences of length 20 because this was the maximum number of samples needed for an agent to optimally solve any of the maze configurations, starting in any initial position and going to one of the possible goals.

For the second test, we repeated the same process used for the first test but changed how the samples were gathered. In this second test, instead of pre-sampling 250 initial positions and asking for a sequence of decisions between each position and each goal, we pre-sampled 250 sets of 20 random states each for each of the 4 world configurations. Then, for each set we got, for each state and testing condition, the prescribed action for each possible goal. Thus, we got 250 sets of 20 state-action pairs for each possible goal and maze configuration. Finally, we repeated the same procedure as in the first test of giving samples to a learner and obtaining its goal inference.

5.3.2. Results

After running the two tests, we aggregated the results for each test by testing condition and number of samples given to the learner, obtaining the average ratio of correct predictions.

The results for the first test can be observed in Fig. 8. The results show that both approaches allowed the agent to learn at a similar rate, although the samples from a PoLMDP policy performed slightly better at making the learning agent understand the underlying reward function. However, these results do not show a significant difference between the two approaches when the examples used are related to each other.

The results for the second test can be observed in Fig. 9. The results show that, after the 20 examples, both learning agents were capable of consistently identifying the teacher's reward function. However, the two approaches show significant differences in performance: when using examples from the PoLMDP policy, the learner identifies the target task significantly faster than when using examples from the optimal policy. For example, when using a PoLMDP policy, the learner achieves a correct prediction ratio of 80% after only 5 shown examples. The same performance was only achieved after observing 7 examples, from the optimal policy. Further analysis of the results shows that the performances of the two approaches only become similar after 16 samples, with the performance of the PoLMDP remaining better than the optimal MDP.

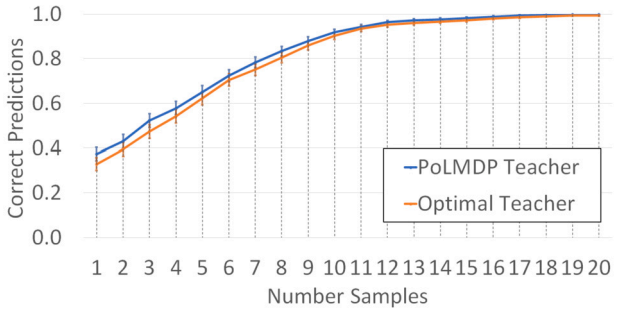


Fig. 8. Ratio of correct predictions depending on the number of examples shown to an IRL agent, with error bars. These results pertain to the condition where the samples formed a complete sequence of decisions.

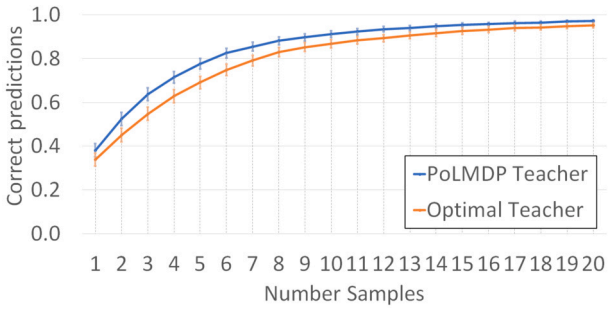


Fig. 9. Ratio of correct predictions depending on the number of examples shown to an IRL agent, with error bars. These results pertain to the condition where the samples had no specific correlation between each other.

5.3.3. Discussion

In the first test, despite the legible decision-making policies allowing for slightly better learning performances than the standard optimal MDP policies, the differences between the two approaches were not significant. Since the samples are sequentially related to one another, both approaches achieve similar overall performance. However, if we observe the early stages of the learning process, we can observe less overlap between the two approaches, hinting at the possibility that, with fewer samples, PoLMDP policies provide more information about the teacher's underlying reward function.

In the second test, we observed a significant difference between the two approaches. PoLMDP samples lead to an improvement of the rate of prediction of the teacher's objective by the learning agent. With less than 10 samples shown to the learner, the learner paired with the PoLMDP got an improvement of 5%–10% in identifying the teacher's model. Thus, we observe that PoLMDP can better convey the intention of the goal to achieve in cases where the observed behaviour is not guaranteed to be logically connected.

It is also interesting to note the several differences between Figs. 8 and 9. In Fig. 8, the learner is slower at the beginning but eventually reaches 100% performance. In Fig. 9, on the other hand, we can see that the learner starts faster but takes longer to attain 100% accuracy. This can be understood by considering the data provided to the learner. In Fig. 8, the data consists of trajectories ending up in the goal state. While "exploring" less the environment, they eventually provide information that provides exact identification of the goal. In Fig. 9, since the data consists of random samples, provides better coverage of the state space (thus the faster converging at the beginning) but may never completely disambiguate what the goal is.

Overall, the results of the evaluation of the performance of both approaches in terms of the information they provide about the task, show that using PoLMDP generated samples leads to an improved performance by the learner and are, thus, more informative. The improvement is mainly observed when the learner observes samples that do not necessarily correspond to complete trajectories. However, even when the samples do correspond to trajectories, the learner agent with the PoLMDP teacher was able to generally outperform the learner agent paired with an optimal teacher. These results establish that the policy provided by PoLMDP is, indeed, more informative regarding the task that the agent is trying to solve.

5.4. User study

The user study is an important aspect of our evaluation because this framework is meant to improve the interaction between robots or other autonomous agents and humans. Thus, to correctly infer the impact PoLMDP has on possible human users, we evaluate if a robot using a PoLMDP policy is better at conveying intentions than a robot using a policy that maximizes the underlying reward function.

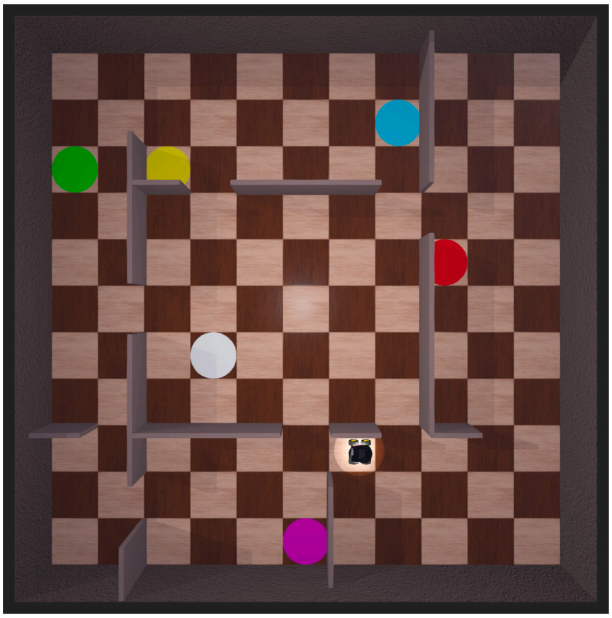


Fig. 10. Still of a possible video the participants would watch. Each of the coloured circles is a possible location the robot could move towards and the participants had to predict the correct one as soon as possible.

5.4.1. Setup

We conducted a user study on the Prolific[1] platform, an online platform to conduct academic research and data collection. Our study compares the performance of our PoLMDP policy, against an optimal policy, in conveying a robot's internal goals. We explore the question:

"Does the PoLMDP generate policies that lead to a more informative robot decision-making?"

To support our exploration of the problem and answer the research question, we postulate the following working hypotheses:

H1 *Participants will understand better the robot's intentions, when paired with a PoLMDP policy than when paired with an optimal policy.*
H2 *Participants will understand quicker and more confidently the robot's intentions when paired with a PoLMDP policy than when paired with an optimal policy.*

We designed an online study, disguised as a guessing game, where the participants had to correctly predict where the robot was moving towards. The conditions of our study were the type of policy used by the virtual robot: the robot either used a PoLMDP policy or an optimal policy to reach the goal. We followed a between-subjects design, meaning that each participant was exposed only to one of the study conditions, *i.e.*, a participant either interacted only with a virtual robot using an optimal policy or a virtual robot using a PoLMDP policy.

In the study, each participant observed 10 small videos of a robot moving to one of 6 differently coloured areas, in a maze world scenario. Fig. 10 shows a still from one of the possible videos the participants would watch. The 10 videos were randomly sampled from a pool of 37 possible videos, with each video generated using the WeBots[2] simulator. For the trajectories of the robot in each video, we sampled the robot's starting position, the robot's goal, and one world configuration out of four possible. For the sampling, we kept a similar distribution of world configurations and goals. Having multiple configurations avoided participants from learning the layout to try and predict the robot's goal and were designed to have similar difficulty in predicting the goals.

The participants had to correctly predict which of the coloured areas was the robot's objective and as fast as possible. For each video, the participants had a play and stop button to control how much of the video to watch. Each participant could stop and restart the video as many times as the participant wanted until they felt they knew the objective. The stop button was disabled for the first 5 seconds of the video, in order to promote participants to watch the video and not only randomly guess. However, after making the prediction, the participant could not go back on the prediction. After making the prediction, the participant also rated how confident they were in the prediction.

In order to encourage faster predictions, each participant's answer was scored from 0 to 10: scoring a 0 for each wrong prediction and then an increasingly higher score, starting at 1 point and going to a maximum of 10 points, the faster they correctly predicted

[1] https://prolific.co/.
[2] https://cyberbotics.com/.

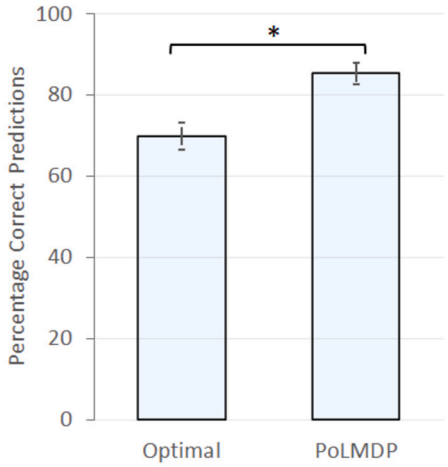


Fig. 11. Percentage, with 95% confidence interval error bars, of participants that correctly predicted the robot's objectives, according to the type of policy used. (*$p < 0.05$).

the robot's objective. At the end of the survey, the participants were presented with their final score and a table that showed each prediction and the corresponding correct prediction.

Before beginning the study, the participants had to read a consent form and choose if they consented to the study. After that, they answered some demographic questions about their age, gender, nationality, degree of education, occupation, and familiarity with robots.

5.4.2. Results

We recruited 150 participants through the Prolific platform, with 66% from Europe, 19% from Africa, 11% from Northern America and Mexico, and the remaining 3% from South America and the Middle East. The participants' average age was between 18 and 29 years old, but their ages varied from 18 to 70 years old. Their gender distribution was 59% female, 38% male, and less than 3% identified as non-binary. Finally, 88% reported having had little or no interaction with robots in their lives, 9% reported interacting occasionally with robots, and less than 3% interact frequently with robots.

Besides the demographic data, for each participant, we measured the average time taken to predict the robot's objective and the percentage of correct predictions. We also measured the self-disclosed rating of confidence in the predictions. Since each participant answered 10 different videos and the videos were randomly sampled from a pool of 37 videos for each condition, we measured the participants' answers for each video presented.

To measure the average time taken to predict the robot's objective, we recorded the time stamp on the video when the participant stopped the video before making a prediction and the full length of the video. Then, if a participant correctly predicted the robot's objective, the participant's time would be the recorded timestamp, if the prediction was wrong the prediction time would be the full length of the video. Regarding the rating in confidence, each participant was asked to rate their confidence on a 7-point Likert scale after making a prediction in each video.

Before we started the analysis of the results, we conducted a normality test on the three measures used. This test reported that the answers obtained did not follow a normal distribution, so all analyses use non-parametric tests or tests that do not assume normality of answers.

The analysis of the number of correct predictions yielded that participants paired with PoLMDP correctly predict the robot's objective in 85% of the predictions, while those paired with the optimal condition correctly predicted the objective in 70% of the predictions. We conducted a Pearson's Chi-Square test, $\chi^2(1) = 48.864, p < 0.001$, which showed that the difference in percentages was significant, supporting our **H1** hypothesis. Fig. 11 shows the results for the percentage of correct answers according to the type of policy used.

The analysis of the average time to correctly predict the objective showed that on average participants paired with the PoLMDP condition took 15.67 seconds to correctly predict the robot's objective, while participants paired with the optimal condition took 18.22 seconds. The difference between the conditions, shows a reduction of 14% of time to predict the robot's objective, in participants paired with PoLMDP. We conducted a Mann-Whitney test that showed the difference was significant, $U = 177976, p < 0.001$, thus supporting our hypothesis **H2**. Fig. 12 shows the results for the average time to correctly predict the robot's objective, with the standard error bars.

Regarding the self-rated confidence in the predictions, Fig. 13 shows a boxplot comparing the two conditions. Participants paired with the PoLMDP condition rated their confidence, on average, as 6.08 out of 7 while participants paired with the optimal condition rated as 5.78 out of 7. A Mann-Whitney test conducted compared the two conditions averages, $U = 231203, p = 0.001$, showing that participants paired with the PoLMDP condition were statistically more confident in their predictions than those paired with the optimal condition. These results support hypothesis **H2**.

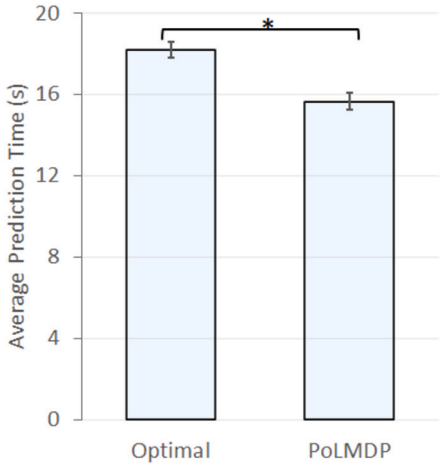


Fig. 12. Average time, with 95% confidence interval error bars, to correctly predict the robot's objective, according to the type of policy used. ($*p < 0.05$).

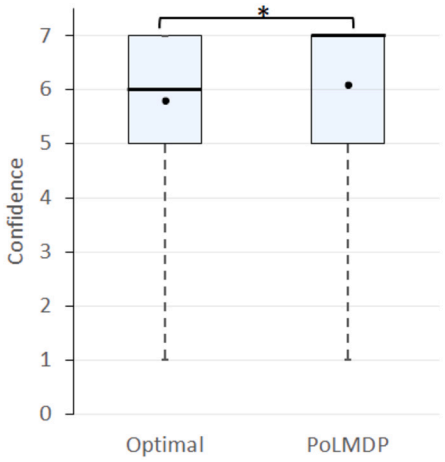


Fig. 13. Boxplot comparing the self-rated confidences in the predictions, according to the type of policy used. The average for each policy is marked with a dot and a thicker black line marks the median. ($*p < 0.05$).

After the initial analysis, we conducted a follow-up analysis exploring the 88% of participants that reported little or no interaction with robots. This analysis is interesting because it can provide insights specific to human-robot interactions with humans that have had little exposure to robots and so they are not as used to look at communication signals in robots as humans with more exposure are used.

In this follow-up analysis, we again looked at the percentage of correct predictions, response time, and self-rated confidence between the two conditions. Before doing the statistical analysis, we conducted a normality test on this subset of results for the participants with little or no robot interaction and, again, the normality test reported that the results did not follow a normal distribution. With this in mind, we conducted non-parametric tests on the three metrics for the subset of participant answers.

Regarding the percentage of correct predictions, we conducted a Pearson's Chi-Square test comparing the percentages of correct predictions between optimal and PoLMDP policies. The Chi-Square test showed a significant difference in percentages, $\chi^2(1) = 34.998, p < 0.001$, with PoLMDP policies leading to a higher percentage of correct predictions—85% of correct predictions when paired with PoLMDP against 71% of correct predictions when paired with optimal policies. The results of the Chi-Square test are aligned with our **H1** hypothesis. Fig. 14 shows a bar plot comparing the percentage of correct predictions with the 95% confidence interval.

To compare the average time taken to correctly predict the robot's objective, we conducted a Mann-Whitney test. The Mann-Whitney test showed a significant difference between the two conditions, $U = 96185.5, p < 0.001$, with the participants paired with PoLMDP policies taking an average of 14 seconds to correctly predict the robot's objective, while the participants paired with optimal policies took on average 15.52 seconds. Thus, participants paired with PoLMDP took on average 1.5 seconds less than those paired with optimal policies, a reduction of 10% of prediction time, which is aligned with our **H2** hypothesis. Fig. 15 shows a bar plot comparison between the average time to correctly predict the robot's objective, with a 95% confidence interval, for the participants with little or no interaction with robots.

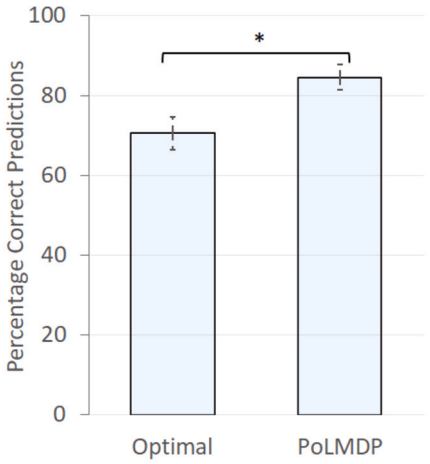


Fig. 14. Results, for the participants with little or no interaction with robots, of the percentage of correct predictions, with 95% confidence interval, according to the type of policy used. ($*p < 0.05$).

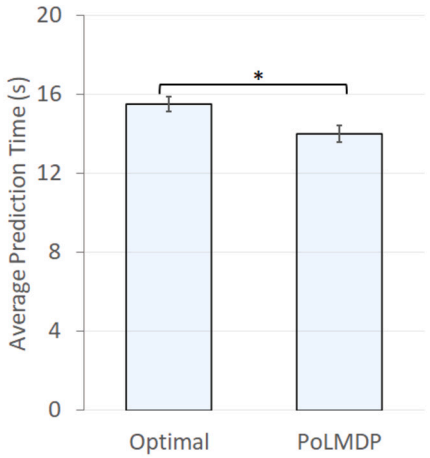


Fig. 15. Results, for the participants with little or no interaction with robots, of the average time to correctly predict the robot's objective, with 95% confidence interval error bars, according to the type of policy used. ($*p < 0.05$).

Finally, regarding the comparison of self-rated confidence in the participants with little or no previous interaction with robots, we conducted a Mann-Whitney test to understand if there were differences between the conditions. The Mann-Whitney test did not show a significant difference between the two conditions, $U = 122114, p = 0.334$, with participants paired with both conditions reporting an average confidence of 6 on a 7-point Likert scale. Thus, in the subset of participants that have had little exposure to robots, the two conditions have not had a different impact on the participant's confidence in the predictions. Fig. 16 shows a boxplot comparing the results of the self-rated confidence between the two conditions, for the participants with little exposure to robots.

Given the difference of 15% in correct predictions between the two conditions, we conducted an extra analysis to verify the existence of false confidence in the participants that incorrectly predicted the robot's objective. This analysis led to us finding no significant differences between participants in the different conditions, with participants in both conditions taking an average of 8 seconds to issue a prediction and reporting a confidence of 5 out 7 in the self confidence question. So, despite the stark difference between conditions when the participants correctly predict the robot's intentions, when participants wrongly predict there are no differences between the two conditions. This leads to us concluding that, despite legible decisions leading to higher prediction confidence when humans correctly predict the robot's objectives, it does not appear to cause false confidence when the prediction is incorrect.

5.4.3. Discussion

The results of the user study support both of our working hypotheses, thus showing the positive impact of using our PoLMDP policy. Our approach allowed the robot to be more expressive regarding its internal goals, making it clearer for the users interacting with the robot what the robot was trying to achieve.

Participants paired with our legible policy saw an increase of 15% in correct predictions, taking on average 3 seconds less to predict the robot's objective. These two aspects support the usefulness of this type of policy in interaction scenarios between humans

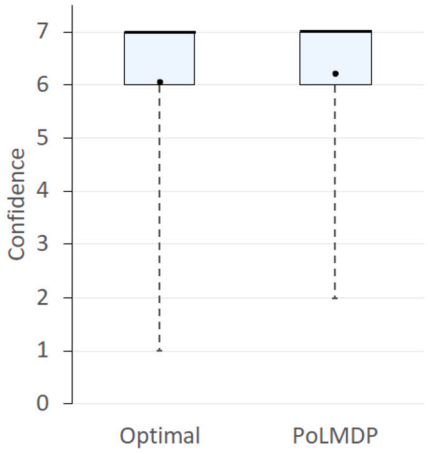


Fig. 16. Boxplot comparing the self-rated confidences in the predictions, for the participants with little or no interaction with robots, according to the type of policy used. The average for each policy is marked with a dot and a thicker black line marks the median.

and robots: by causing humans to have better predictions about a robot's intentions and with less time needed to predict. Allowing humans to understand the agent's objective faster may lead to more efficient interactions, thus improving the role of autonomous agents in society.

Another aspect to highlight from this study is the higher confidence human participants felt in their predictions when correctly predicting the robot's objective, without causing false confidence when the predictions were incorrect. The confidence was not only higher but consistently higher in humans paired with PoLMDP as seen by the boxplot in Fig. 13 where the median line is at the highest possible value. The existence of high confidence, without causing false confidence in situations of incorrect predictions, is important because it allows humans to better focus on their personal objectives and tasks and, in collaborative scenarios, this can lead to increased task performance. Thus, an increase in confidence has a great impact on the success of an interaction and the overall interaction experience.

One final interesting insight from the user study comes from analysing the results of the participants with little or no prior interaction with robots. After analysing the results for that set of participants, we found that in participants without much prior exposure to robots the kind of policy used did not affect much the participants' confidence in their predictions; however, participants paired with PoLMDP policies still predicted the correct object more frequently and faster than those paired with optimal policies. So, despite humans without much prior exposure to robots not being used to interpret robotic behaviours, they still understand robots' intentions faster with legible policies.

6. Conclusion

In this paper, we present PoLMDP, a framework to create legible policies in sequential decision-making problems. PoLMDP generates legible actions, by choosing the ones that, at a given moment, are more representative of the robot's intentions—distancing the robot from other possible objectives.

Through a combination of two computational evaluations and one online user study, we have shown the positive impact of our PoLMDP framework. We show that PoLMDP outperforms the existing competing method (Legible MDPs), generating legible behaviours in less time while allowing agents to be more efficient. We also show that the examples given by PoLMDP are better at conveying the task being executed, allowing for an IRL agent to learn the underlying reward functions faster than with examples from the optimal policy. Finally, through an online user study, we show that PoLMDP is better at generating behaviours that convey a robot's intended goal than using optimal policies, giving humans more time to adapt to a system's actions and act accordingly.

In this work we have explored the application of PoLMDP in Markovian scenarios with full observability and known dynamics, such as MazeWorld settings, showing that PoLMDP can achieve performance equiparable to other state-of-art methods for legible decision making, at a fraction of the execution time. However, not all scenarios allow for full observability or for full known dynamics and so it would be interesting, in the future, to explore if our PoLMDP formulation holds in such complex scenarios.

We envision PoLMDP as a promising framework for applications in scenarios of search and rescue operations, where a team must coordinate to explore an area with various points of interest; for collaborative applications in healthcare scenarios where reading actions from a co-worker's body language are essential for team coordination; or for collaboration tasks in home scenarios, where the robot acts in an independent capacity to achieve a set of objectives—like a Roomba cleaning the house.

CRediT authorship contribution statement

Miguel Faria: Formal analysis, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Francisco S. Melo:** Investigation, Methodology, Supervision, Writing – review & editing. **Ana Paiva:** Investigation, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

The authors report financial support was provided by the Portuguese Foundation for Science and Technology.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by national funds through the Portuguese Fundação para a Ciência e a Tecnologia under project UIDB/50021/2020. Miguel Faria acknowledges the PhD grant PD/BD/143144/2019.

References

[1] Behnoush Abdollahi, et al., Transparency in fair machine learning: the case of explainable recommender systems, in: Human and Machine Learning, Springer, 2018, pp. 21–35.
[2] Rachid Alami, et al., Safe and dependable physical human-robot interaction in anthropic domains: state of the art and challenges, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2006.
[3] Victoria Alonso, et al., System transparency in shared autonomy: a mini review, Front. Neurorobot. 12 (November 2018) 1–11, https://doi.org/10.3389/fnbot.2018.00083, ISSN 16625218.
[4] Leonardo Amado, et al., Goal recognition as reinforcement learning, Proc. AAAI Conf. Artif. Intell. 36 (9) (2022) 9644–9651.
[5] S. Anjomshoae, et al., Explainable agents and robots: results from a systematic literature review, in: International Conference on Autonomous Agents and Multiagent Systems, AAMAS, AAMAS, 2019.
[6] Alejandro Barredo Arrieta, et al., Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.
[7] Chris L. Baker, et al., Action understanding as inverse planning, Cognition 113 (3) (2009) 329–349.
[8] Cynthia Breazeal, et al., Effects of nonverbal communication on efficiency and robustness in human-robot teamwork, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2005.
[9] B. Busch, et al., Learning legible motion from human–robot interactions, Int. J. Soc. Robot. 9 (5) (2017) 765–779, Springer.
[10] Diogo V. Carvalho, et al., Machine learning interpretability: a survey on methods and metrics, Electronics 8 (8) (2019) 832.
[11] Tathagata Chakraborti, et al., Balancing explicability and explanation in human-aware planning, arXiv preprint, arXiv:1708.00543, 2017.
[12] Tathagata Chakraborti, et al., Explicability? Legibility? Predictability? Transparency? Privacy? Security? The emerging landscape of interpretable agent behavior, in: Proceedings of the International Conference on Automated Planning and Scheduling, ICAPS, vol. 29, 2019, pp. 86–96.
[13] Y. Che, et al., Efficient and trustworthy social navigation via explicit and implicit robot–human communication, IEEE Trans. Robot. 36 (3) (2020) 692–707.
[14] Stephen A. Cook, The complexity of theorem-proving procedures, in: Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71, Association for Computing Machinery, Shaker Heights, Ohio, USA, ISBN 9781450374644, 1971, pp. 151–158.
[15] Filipa Correia, et al., A social robot as a card game player, in: Conference: Artificial Intelligence and Interactive Digital Entertainment, AIIDE, 2017.
[16] Finale Doshi-Velez, et al., Towards a rigorous science of interpretable machine learning, arXiv preprint, arXiv:1702.08608, 2017.
[17] A. Dragan, et al., Generating legible motion, in: Robotics: Science and Systems, 2013.
[18] A. Dragan, et al., Legibility and predictability of robot motion, in: ACM/IEEE Int. Conf. Human-Robot Interaction, IEEE, 2013, pp. 301–308.
[19] A. Dragan, et al., Effects of robot motion on human-robot collaboration, in: ACM/IEEE Int. Conf. Human-Robot Interaction, IEEE, 2015, pp. 51–58.
[20] Utkarsh Dwivedi, Introducing children to machine learning through machine teaching, in: Interaction Design and Children, 2021, pp. 641–643.
[21] M. Faria, et al., "Me and you together" movement impact in multi-user collaboration tasks, in: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2017, pp. 2793–2798.
[22] Miguel Faria, et al., Follow me: communicating intentions with a spherical robot, in: IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN, IEEE, 2016.
[23] Miguel Faria, et al., Understanding robots: making robots more legible in multi-party interactions, in: 2021 30th IEEE International Conference on Robot Human Interactive Communication, RO-MAN, 2021, pp. 1031–1036.
[24] Jaime F. Fisac, et al., Generating plans that predict themselves, in: Algorithmic Foundations of Robotics XII, Springer, Cham, 2020, pp. 144–159.
[25] Katie Genter, et al., Ad hoc teamwork for leading a flock, in: International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS, 2013, pp. 531–538.
[26] N. Gildert, et al., The need for combining implicit and explicit communication in cooperative robotic systems, Frontiers (2018) 65.
[27] Piotr J. Gmytrasiewicz, et al., A framework for sequential planning in multi-agent settings, J. Artif. Intell. Res. 24 (2005) 49–79.
[28] Soheil Habibian, et al., Encouraging human interaction with robot teams: legible and fair subtask allocations, IEEE Robot. Autom. Lett. 7 (3) (2022) 6685–6692, https://doi.org/10.1109/LRA.2022.3174264.
[29] Mark K. Ho, et al., Showing versus doing: teaching by demonstration, in: Advances in Neural Information Processing Systems, NIPS, vol. 29, 2016.
[30] Chien-Ming Huang, et al., Anticipatory robot control for efficient human-robot collaboration, in: ACM/IEEE International Conference on Human-Robot Interaction, HRI, IEEE, 2016, pp. 83–90.
[31] Sandy H. Huang, et al., Enabling robots to communicate their objectives, Auton. Robots 43 (2) (2019) 309–326.
[32] Leslie Pack Kaelbling, et al., Planning and acting in partially observable stochastic domains, Artif. Intell. 101 (1–2) (1998) 99–134.
[33] Levente Kocsis, et al., Bandit based Monte-Carlo planning, in: European Conference on Machine Learning, Springer, 2006, pp. 282–293.
[34] M. Kwon, et al., Expressing robot incapability, in: ACM/IEEE Int. Conf. Human-Robot Interaction, 2018, pp. 87–95.
[35] Songpo Li, et al., Implicit intention communication in human–robot interaction through visual behavior studies, IEEE Trans. Human-Mach. Syst. 47 (4) (2017) 437–448.
[36] M. Littman, et al., The computational complexity of probabilistic planning, J. Artif. Intell. Res. 9 (1998) 1–36.
[37] M. Lopes, et al., Active learning for reward estimation in inverse reinforcement learning, in: Proc. Eur. Conf. Machine Learning and Knowledge Discovery in Databases, 2009, pp. 31–46.

[38] Aleck M. MacNally, et al., Action selection for transparent planning, in: International Conference on Autonomous Agents and Multiagent Systems, AAMAS, 2018, pp. 1327–1335.
[39] O. Madani, et al., On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems, in: Proc. 16th AAAI Conf. Artificial Intelligence, 1999, pp. 541–548.
[40] C. Mavrogiannis, et al., Social momentum: a framework for legible navigation in dynamic multi-agent environments, in: ACM/IEEE Int. Conf. Human-Robot Interaction, 2018.
[41] Reuth Mirsky, et al., A survey of ad hoc teamwork research, in: European Conference on Multi-Agent Systems, Springer, 2022, pp. 275–293.
[42] Shuwa Miura, et al., A unifying framework for observer-aware planning and its complexity, in: Uncertainty in Artificial Intelligence, PMLR, 2021, pp. 610–620.
[43] Shuwa Miura, et al., Maximizing legibility in stochastic environments, in: 2021 30th IEEE International Conference on Robot & Human Interactive Communication, RO-MAN, IEEE, 2021, pp. 1053–1059.
[44] A. Ng, et al., Algorithms for inverse reinforcement learning, in: Proc. 17th Int. Conf. Machine Learning, 2000, pp. 663–670.
[45] S. Nikolaidis, et al., Viewpoint-based legibility optimization, in: ACM/IEEE Int. Conf. Human-Robot Interaction, 2016, pp. 271–278.
[46] C. Papadimitriou, et al., The complexity of Markov decision processes, Math. Oper. Res. 12 (3) (1987) 441–450.
[47] Raul Paradeda, et al., The role of assertiveness in a storytelling game with persuasive robotic non-player characters, in: Annual Symposium on Computer-Human Interaction in Play, 2019, pp. 453–465.
[48] Karl Popper, The myth of the framework, in: Rational Changes in Science, Springer, 1976, pp. 35–62.
[49] D. Ramachandran, et al., Bayesian inverse reinforcement learning, in: Proc. 20th Int. Joint Conf. Artificial Intelligence, 2007, pp. 2586–2591.
[50] Maha Salem, et al., Would you trust a (faulty) robot?: effects of error, task type and personality on human-robot cooperation and trust, in: ACM/IEEE International Conference on Human-Robot Interaction, HRI, ISBN 9781450328838, 2015, pp. 141–148.
[51] Wojciech Samek, et al., Explaining deep neural networks and beyond: a review of methods and applications, Proc. IEEE 109 (3) (2021) 247–278, https://doi.org/10.1109/JPROC.2021.3060483.
[52] S. Saunderson, et al., How robots influence humans: a survey of nonverbal communication in social human–robot interaction, Int. J. Soc. Robot. 11 (4) (2019) 575–608, Springer.
[53] D.J. Strouse, et al., Learning to share and hide intentions using information regularization, in: Advances in Neural Information Processing Systems, NIPS, vol. 31, 2018.
[54] Freek Stulp, et al., Facilitating intention prediction for humans by optimizing robot motions, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2015, pp. 1249–1255.
[55] Sebastian Wallkötter, et al., SLOT-V: supervised learning of observer models for legible robot motion planning in manipulation, in: IEEE International Conference on Robot and Human Interactive Communication, RO-MAN, IEEE, 2022, pp. 1421–1428.
[56] Richard Warner, et al., Making artificial intelligence transparent: fairness and the problem of proxy variables, Crim. Justice Ethics 40 (1) (2021) 23–39.
[57] Xiaojin Zhu, Machine teaching: an inverse problem to machine learning and an approach toward optimal education, Proc. AAAI Conf. Artif. Intell. 29 (1) (2015).
[58] Xiaojin Zhu, et al., An overview of machine teaching, arXiv:1801.05927 [abs], 2018.
[59] Brian D. Ziebart, et al., Maximum entropy inverse reinforcement learning, in: AAAI Conference on Artificial Intelligence, Chicago, IL, USA, vol. 8, 2008, pp. 1433–1438.