Neural Computing and Applications (2023) 35:16781–16790
https://doi.org/10.1007/s00521-022-07942-7

S.I.: HUMAN-ALIGNED REINFORCEMENT LEARNING FOR AUTONOMOUS AGENTS AND ROBOTS

Policy regularization for legible behavior

Michele Persiani[1] · Thomas Hellström[1]

Received: 9 December 2021 / Accepted: 11 October 2022 / Published online: 26 October 2022

Abstract
In this paper we propose a method to augment a Reinforcement Learning agent with legibility. This method is inspired by the literature in Explainable Planning and allows to regularize the agent's policy after training, and without requiring to modify its learning algorithm. This is achieved by evaluating how the agent's optimal policy may produce observations that would make an observer model to infer a wrong policy. In our formulation, the decision boundary introduced by legibility impacts the states in which the agent's policy returns an action that is non-legible because having high likelihood also in other policies. In these cases, a trade-off between such action, and legible/sub-optimal action is made. We tested our method in a grid-world environment highlighting how legibility impacts the agent's optimal policy, and gathered both quantitative and qualitative results. In addition, we discuss how the proposed regularization generalizes over methods functioning with goal-driven policies, because applicable to general policies of which goal-driven policies are a special case.

Keywords Reinforcement learning · Transparency · Interpretability · Legibility

1 Introduction

As widely agreed in Explainable Artificial Intelligence, well-functioning collaboration between humans and artificial agents requires transparency [1]. Agents should not only perform their assigned tasks efficiently and accurately, but should also make sure that the humans in their operative context understand their intentions and actions.

Facilitating intention recognition through a behavior that is understandable by a human observer has several advantages [2]. For example, in human-robot interaction signaling the robot's intention increases collaborators' trust in the robot, safety, and fluency of interactions, because aiding collaborators to predict what the robot is doing or will do [3–5], and in conditions of shared control allows to mediate, arbitrate, and guide the interaction [6] by informing the user about the robot's intended action. In applications for autonomous vehicles simple solutions augmenting the driver understanding of the car's intentional state, like sharing its goal, is sufficient to increase trustworthiness and acceptability of the autonomous driving system, as well as acceptance of higher levels of automation [7]. In addition, recent developments in technologies for virtual or mixed reality are further enabling and enhancing methods for intentionality in physical robots, by allowing to plot and manipulate the robots' intentional states in the virtual 3D world [8].

Given the importance of intentions during interactions with artificial agents, it is therefore becoming relevant to combine methods that allow to express intentions with techniques generating highly performing behavior. The online creation of behavior of which intention is easily discernable or that is furnished with congruent explanations is addressed in Explainable Planning under the umbrella of *interpretable behavior*, where several methods to regularize behavior for explicability [9], predictability [10] or legibility [11, 12] have been proposed. These techniques relate to an implicit communication of intention

✉ Michele Persiani
michelep@cs.umu.se

Thomas Hellström
thomas.hellstrom@umu.se

[1] Department of Computing Science, Umeå University, Umeå, Sweden

by making it transparent to its user, and is in contrast with explanations that instead is an explicit communication. Transparency is achieved by interacting with a user observer model. For example, legibility skews plan trajectories such that their goal is easily discernable, explicability makes sure that observations have at least one-associated complete plan, or predictability reduces the amount of possible future possible trajectories.

While a substantial amount of formalizations of interpretable behavior exists in the Explainable Planning literature, there is very little-related work for the framework of Reinforcement Learning (RL). RL has been shown to produce powerful agents for a variety of domains (including robotics, games or recommender systems) often surpassing human performance, however, the RL framework still lacks formalization about creating agents that are interpretable as intended in Explainable Planning, and mostly borrows its definition of interpretability from the Machine Learning (ML) literature. This definition is more concerned into making the decision taken by the algorithm explainable by a domain expert upon inspection in an offline setting, rather than to enable interpretability online during collaborations, therefore resulting unsuitable in fulfilling the needs of transparency of online interactions. There is therefore still a large untapped potential in adapting methods for interpretability to RL. This would also provide valuable input for research in explainability that at the moment contemplates advanced methods such as those based on neural networks mostly as black boxes generating behavior that is optimal yet highly uninterpretable from a human perspective [13].

With the goal of including legibility criteria in RL, in this paper we translate legibility from Explanable Planning to the RL framework as a measure of discernability of policy, that we loosely equal to the agent's intention. As we propose, injecting legibility inside an agent's policy doesn't require to modify components of the learning algorithm. We rather suggest to evaluate how the optimal policy may produce state-action pairs that would make an observer model to infer a wrong policy, and to later find a trade-off that minimizes those while remaining consistent to the original policy. This is performed through what we refer to as the *Mirror Agent Model* that is a model furnishing legibility to the agent *as a service* [14], that is without modifying its underlying functioning or training procedure. As we will later discuss, this setting adds several degrees of freedom to the previously proposed methods from the literature relying on augmenting the agent's training (such as regularizing its reward function).

2 Background

Since RL borrows the term "interpretability" mostly from the ML literature [15, 16], merging the terminology from Explainable Planning and Reinforcement Learning could create some confusion. In ML interpretability generally means to provide insight into the agent's mechanisms such that its decisions are understandable by an expert upon inspection [16]. This can be achieved firstly by translating the classifiers' latent features responsible for its decisions into a space that is interpretable, and then compute explanations on that space [17]. In RL, [18] for example proposes to use attention to visualize which features the deep Q-network attends when taking decisions, while [19] trains linear tree models on Deep Q-networks to obtain corresponding interpretable models. See [15] for a survey of this type of techniques applied to RL.

These techniques for interpretability have been shown useful in many ML application domains by giving insight into models' decisions. They have, for example, been successful in health-care [20], and societal (e.g., decisions regarding loans, hiring, risks, etc.) applications. However, they may be less suitable in domains characterized by real-time interaction, such as in human-robot interaction, where the fluency of the interaction prohibits deep inspections of the decision-making algorithm. Also, while the produced explanations in terms of relevant features could be understood by an expert, they may be unsuitable for users who are uninformed of the underlying models, and more focused on common sense reasoning. People generally very good at forming hypotheses on intentions and beliefs explaining an observed behavior through what is referred to a theory of mind reasoning [21]. However, it has been commonly shown how the behavior of advanced agents operating at human level, such as in competitive games, are often beyond human intuition and highly inexplicable [22, 23]. Especially for such cases, but also in general, it is therefore necessary to regularize artificial agents toward behaviors compatible with common sense reasoning, while maintaining their high performance.

In this paper we refer to interpretability as intended in planning, where an agent behavior is interpretable when an observer can easily discern what the agent is doing by understanding its intention [24]. Also when applied to RL, this definition conforms better to real-time interaction in the presence of an observer that could be either passive or part of a larger collaborating agent. As previously introduced, in this context a multitude of definitions capturing smaller aspects of interpretability have been used. Each aspect expresses different types of expectations that an eventual observer has on the agent, such as expectations about its goal [11], expectations about entire future

trajectories [10], or expectations toward a communication model [25]. While there is a lot of variety in the models and theories leveraged by all this techniques, it can be generally shown that this set of methods requires an expectation model that is a second-order theory of mind focused on the observer's inferences about the agent [2, 24], and that interpretable behavior can be seen as minimizing the distance between the estimated model possessed by the observer and the true model of the agent (see Fig. 1). The agent's behavior is interpretable whenever conforming with the expectations casted by the second-order model, and uninterpretable when not conforming [2].

In agents applications the second-order theory of mind is the model that the agent thinks the observer is using to interpret its behavior and can have many forms, for example, in [10] it is a label predicting whether a human observer is understanding the agent, while in [26] is a complete planning model. In general, simple observer models are easier to maintain aligned with the actual expectations of the user, while those that are more structured allow to simulate with greater detail the inferences of the observer. Also, structured models can be selectively changed through a reconciliation process [26] thus ultimately allowing the agent to autonomously re-align its model with the observer's whenever it detects the need.

To the best of our knowledge very little work exists in RL relating to interpretable behavior as we just described. Both [27, 28] propose methods relying on a transposition of the original formulation of legibility. The methods result applicable only for goal-driven policies, thus excluding all other types of policies available in various RL frameworks. In addition, they require to specify distance measures between states that, while easy for manipulators working in the cartesian space, can be a difficult task for arbitrary state-spaces.

Rather than relying of goal locations, we define a legibility criteria that is directly applicable on policies. A regularization method similar to ours is proposed in works on offline policy learning [29–31] where during training the agent's on-policy behavior is regularized toward another behavior. We can see our method as a specific application inside this class of methods, where the policy is regularized toward the legible policy. However, we propose to regularize the policy after training, while the training of the agent is left untouched. The proposed method furnishes explainability as a service, that is as a wrapper of an existing computational model. This has major advantages in terms of usability of the methods because doesn't require to retrain the agent for every variation of the observer model.

3 Method

The main goal of interpretable behavior is to bring the intention predicted by the observer's model close to the intention of the agent, and to maintain such closeness in time. Consistently with the definition of a legible intention we define a legible policy as: *An agent's policy is legible if it is discernible from a set of other policies*. It is useful to work with this definition because it reflects the general case where an observer is attempting to understand which policy the agent is currently enacting among a set of candidates. Furthermore, the definition doesn't pose constraint on the type of policy but can be applied to arbitrary policies. The goal of legibility is therefore to help the observer to infer the correct policy from the set of those being considered. For this case we hypothesize an observer watching the agent and inferring the policy it is currently pursuing.

The agent can simulate the presence of an observer by implementing a second-order theory of mind modeling the expectations that it is using to infer intentions. To implement the second-order theory of mind we utilize a middle way between the expressiveness of a complete agent model, and the simplicity of using a hand-crafted solution. This model for theory of mind reasoning, that we refer to as the *Mirror Agent Model*, describes agent and observer models as two equivalent Bayesian networks denoted P_R and P_R^H (Fig. 2). P_R determines how the agent acts, while P_R^H is the observer's model of how the agent acts. Since the real observer model is part of the observer and it is not directly accessible by the agent. The agent must therefore

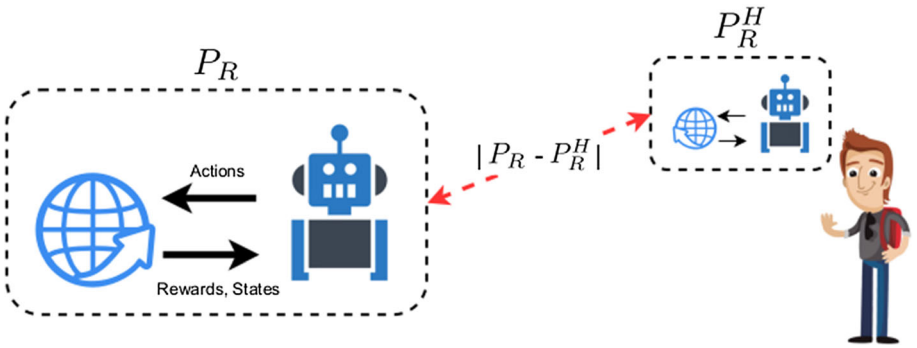

Fig. 1 P_R: an RL agent interacting with its environment. P_R^H: model of the expectations that an observer has about the agent. The goal of interpretable behavior is to keep the distance $|P_R - P_R^H|$ low, signifying that the agent's behavior effectively matches the observer's expectations

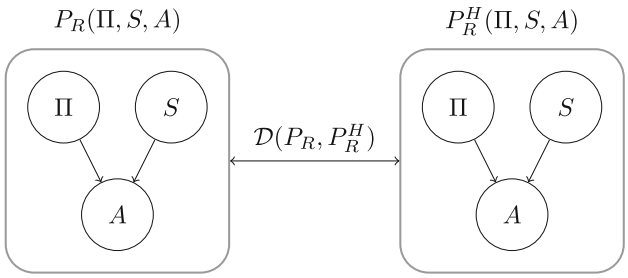

Fig. 2 Agent model and second-order theory of mind as equivalent Bayesian Networks. The networks model how agent and observer respectively select and infer actions using the current state and a set of predefined policies, while the function $\mathcal{D}$ measures the distance between these two processes

for all computations rely on the estimated model P_R^H, the second-order theory of mind. To simplify notations, we make in the following no distinction between these two entities, and we use observer model and second-order theory of mind as interchangeable.

The Bayesian networks are structurally the same and describe the agent as a Markov Decision Process (MDP) with multiple possible policies, however, the random variables (Π, S and A) can be differently distributed in P_R compared to P_R^H, depending on the agent's reasoning and prior information about the observer. A simplifying assumption this model makes is that the user internalizes an agent model with the same structure as the true agent model. While this assumption may not hold in the general case, it can, for example, be achieved by communicating the agent model, or by performing model alignment dialogues with the goal of communicating the latent variables that the agent uses to act.

We assume that the agent has a fixed set of pre-trained policies identified by the random variable $\Pi = \{\pi_0, \ldots, \pi_n\}$. Notably, among these there is the currently pursued policy π_R with $P_R(\Pi = \pi_R) = 1$. Initially, the observer is modelled as ignorant of which policy the agent is pursuing, leading to a uniform prior of the policies: $\forall i\, P_R^H(\pi_i) = k$, $k = \frac{1}{|\Pi|}$. When using Q-learning, two corresponding Q-value tables $Q_R(a, \pi, s)$ and $Q_R^H(a, \pi, s)$ respectively determine the probability distribution for the agent selecting actions, with $P_R(a|\pi, s) = f(Q_R(\pi, s, a))$, and for the observer inferring the agent's actions, with $P_R^H(a|\pi, s) = g(Q_R^H(\pi, s, a))$. The Q-value tables can be obtained using any of the available RL methods, while f and g are arbitrary functions that transform Q-values into probability distributions of actions, for example the Boltzmann or the ϵ-greedy distributions [32].

To be legible, the agent should select actions a that communicate the observer its policy π_R, while avoiding communicating the others. This is obtained by selecting actions based on how they reduce the distance $\mathcal{D}$ between the probability distribution over the agent policies, $P_R(\Pi)$, and the corresponding distribution $P_R^H(\Pi|s, a)$ that the observer infers, given an observation of state-action pair. To implement $\mathcal{D}$ we utilize cross-entropy that specifies how much information would be additionally required to identify π_R by using $P_R^H(\Pi|s, a)$ instead of $P_R(\Pi)$, that is by using the observer model rather than the agent's.

$$\begin{aligned} &\mathcal{D}(P_R(\Pi), P_R^H(\Pi|s, a)) \\ &- \log P_R^H(\pi_R|a, s) \\ &- \log P_R^H(a|\pi_R, s) + \log \mathbb{E}[P_R^H(a|\pi, s)] - \log P_R^H(\pi_R). \end{aligned} \tag{1}$$

where the last line is obtained through the Bayes' theorem and law of total probability. Since the action probabilities in Q-learning depend on the Q-values, we can use Eq. 1 to define regularized versions of the Q-values as:

$$\begin{aligned} &Q_{\text{leg}}(\pi_R, s, a) \\ &Q_R(\pi_R, s, a) - \alpha \mathcal{D}(P_R(\Pi), P_R^H(\Pi|s, a)) \\ &Q_R(\pi_R, s, a) + \alpha \log P_R^H(\pi_R|a, s). \end{aligned} \tag{2}$$

with $\alpha > 0$ determining the magnitude of regularization. In this way, the right part of Eq. 2 regularizes the resulting policy such that the selected actions aim at a small distance between the agent policy and the policy inferred by the observer. Therefore, the decision boundary introduced by legibility impacts the states in which the optimal action $a_{\text{opt}} = \pi_R(s)$ is an action that has high probability also in other policies. In such cases, a trade-off between the optimal action, and a sub-obtimal/legible actions is made.

4 Experiments and evaluation

We tested and evaluated the proposed model with two experiments. The first is an illustrative example in a grid*world setting and is intended to provide insight into how the legible policy modifies the original policy. The second experiment is more extensive and is performed with a Deep Q-network.

4.1 Grid-world experiment

In this experiment we tested the proposed method on a gridworld scenario. The grid is 7x7 and without obstacles. There are 3 possible goals at the corners, for which we trained three corresponding policies with Q-learning. For simplicity we set $Q_R = Q_R^H$ and $f = g$, meaning that the agent assumes the observer to use the same Q-values and derived action probabilities as its own, i.e., $\forall i\, P_R(A|\pi_i, S) = P_R^H(A|\pi_i, S)$. This has the advantage of not require modeling how the observer models the task, which is a costly procedure. However, nothing prohibits usage of

different Q-values for the observer. In such cases, the agent would be evaluated by a different set of policies than those it possesses.

Figure 3 shows in the left column the optimal policies learned by the agent. In the right column the correspondingly legible policies obtained using $\alpha = 1$. The learned policies move toward a wall adjacent the goal, and then approach the goal by walking along the wall. However, to be legible, it is important to approach the right wall that disambiguates the goal location. The legible policies systematically approach an unambiguous wall. Notice also how for g_1, the legible policy makes the agent walk in the middle to avoid approaching the other goals.

4.2 Deep Q-network experiment

In the second experiment we used *OpenAI Gym* [33]. We designed a simulated environment in which the agent had to pass through tunnels of length L and width W, composed of $C + 2$ types of cells: empty cells, obstacle cells, and C types of cells of different colors (see Fig. 4). The agent was defined to see a maximum of S cells in front of it and had 3 possible actions: move one cell up, move one cell down, or stay at the same position. If the agent moves to a colored cell it receives a reward of $+1$ while if it moves to an obstacle it gets a punishment of -10 and a new episode restarts. Moving to an empty cell or to a cell of a color different from its own does not result in any reward or punishment. The environment is not goal-oriented but rather defines regions of reward and of punishment for the agent. These regions can be of arbitrary shape and we used rectangles for colored regions and squares or lines for obstacles.

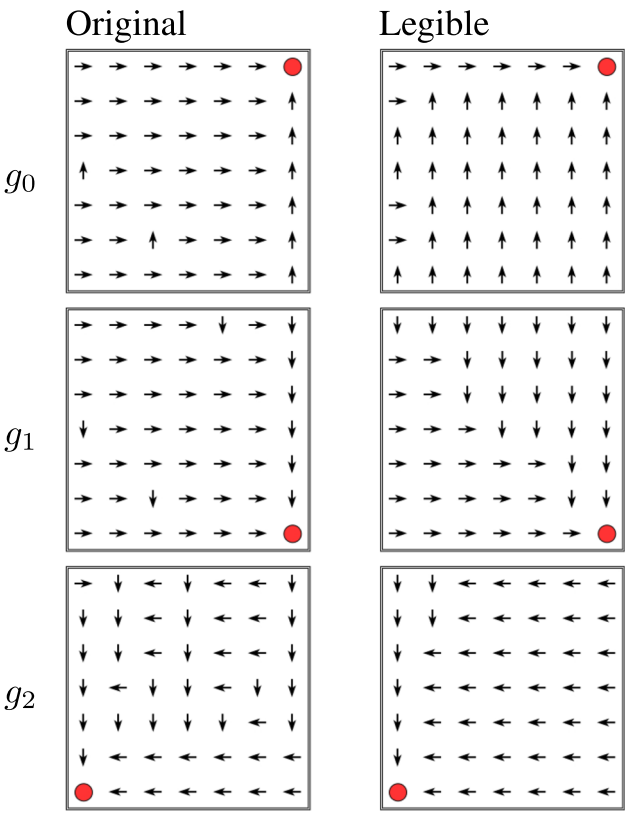

Fig. 3 Left: policies for the three goals (red dots) learned with Q-learning. Right: legible policies. The legible policies avoid ambiguity of goal location

Since the agent is unaffected by cells of a color different from its rewarding color, to simplify the learning process it was trained on tunnels containing only one color and obstacles. Later, tunnels containing C colors are obtained by using C tunnels sharing obstacles and agent position. Inside a single-color tunnel, at every timestep the observation corresponds to a set of three matrices M_0, M_1, M_2 of size $W \cdot S$, each representing a slice of the tunnel up to the agent's sight distance. The first matrix contains only colored cells, the second only obstacles and the third the agent's position. Inside every matrix, each cell is characterized by the summation of three embedding vectors:

$$c_{ij} = w_i + s_j + t_{ij}. \tag{3}$$

where w_i and s_j are position embeddings identifying the cell inside the matrix. For example, $\langle w_0, s_5 \rangle$ indicates cell $0 - 5$. While t_{ij} identifies whether that cell is occupied: in M_0 a cell is occupied if it is colored, in M_1 if it is an obstacle, in M_2 if it contains the agent's position.

Figure 5 shows the employed Q-network. In the network, ϕ is a convolution network which convolves on the matrices of embeddings, and is shared by all the inputs M_0, M_1 and M_2. ψ is a fully connected network that takes as input the vector $\langle \phi(M_0), \phi(M_1), \phi(M_2) \rangle$ and outputs a vector of size 3 for the Q-values.

We trained the agent on 30,000 random, single-color tunnels of length 200 and width 12 cells, while the agent's observation windows was set to 20 cells. For every tunnel 5 colored rectangles and 10 obstacles of shape square or line were randomly placed. Table 1 shows the network's hyperparameters used for training the Q-network.

As previously mentioned, after training to obtain a tunnel with C colors we merged C tunnels at once, with each tunnel containing only cells of the respective color, while all sharing the same obstacles and agent position. In this way, at each step the agent has C different policies to follow, each one seeking a particular color. This is equal to the result of training C different policies simultaneously.

4.2.1 Quantitative evaluation

We tested the proposed method for legible policy in a setting where both agent and observer use the same

Fig. 4 Sampled tunnel environment. While traversing a tunnel the agent is rewarded to walk on cells of its same color (green). Hitting an obstacle (teal) instead punishes the agent and resets the episode

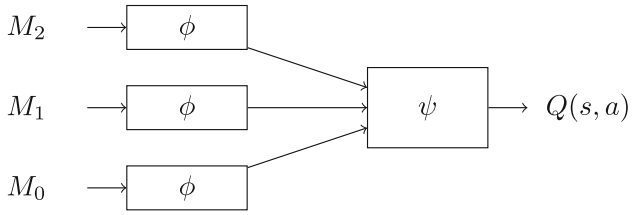

Fig. 5 Q-network for the tunnel enviroment. ϕ: convolution network shared by the three inputs. ψ: fully connected network

Table 1 Hyperparameters of the Q-network for training the agent

Parameter	Amount	Type
Layers_ϕ	(100,100,100,100)	2*D*-convolution
Layers_ψ	(200,200,50)	Fully connected
Embedding size	100	–
Learning rate	$1e-3$	–
Learning rate decay	$1e-4$	–
Buffer size	150,000	–
Policy ϵ	0.1	–
Discount factor	0.98	–

Q-function (the trained Q-network) and the greedy policy to always select the action with highest Q-value. Since the introduced regularization penalizes actions with high probability in other policies, we expected the agent to avoid cells of colors that are not its own. In other words, since the observer model judges the agent's behavior by confronting it with policies that seek cells of given colors, by avoiding cells of other colors the agent decreases the probability of those policies in the observer's inferences.

We tested this hypothesis first quantitatively by measuring the average gathered reward over 200 episodes, while using increasing values for the regularization factor α. Every random tunnel had $C = 4$ colors, 5 rectangular-colored patches for each color, and 10 square obstacles. In this setting we measured the reward gathered by the agent when pursuing the color C_0, and the average reward for the other colors $C_{1..3}$ accumulated while pursuing C_0. We then divided these scores by the maximum rewards that the policy could have gathered for the corresponding colors, thereby obtaining a *reward ratio* with values between 0 and 1. For example, a reward ratio of 0.5 means that the agent accumulated half of the possible maximal reward. As a complement to the reward ratio, *success rate* was computed as the probability of succeeding, i.e., reaching the end of the tunnel without hitting any obstacles during an episode. Table 2 summarizes this experiment.

Table 3 instead summarizes the degree of legibility of the agent's policy measured as the expected probability that the observer model gives to the agent's policy through the episodes:

$$\mathcal{L} = \mathop{\mathrm{E}}_{\langle a,s\rangle \sim \mathcal{E}}[P_R^H(\pi_R|s,a)]. \tag{4}$$

where every state transition is given equal weight $p(s,a) = \frac{1}{|\mathcal{E}|}$. The second row of the table shows the gain of legibility obtained by using the legible policy rather that the original:

$$\mathcal{L}_{\text{gain}}(x) = \frac{\mathcal{L}_{\alpha=x}}{\mathcal{L}_{\alpha=0}}. \tag{5}$$

4.2.2 Qualitative Evaluation

Figure 6 shows the effect of regularization on two sampled tunnels. In the plots red is the rewarding color of the agent and obstacles are in brown. The trajectories in yellow are obtained by simulating and averaging 200 trials. As noticeable the regularized trajectories are sharper and avoid non-red-colored regions. This arguably increases the legibility to a human observer as well. The most convincing reason is that if we would see the agent avoiding e.g., purple cells, we would suppose it is not rewarded by that color. In our experiments this prediction of intention is also

Table 2 Average accumulated reward ration by the policies for color C_0 and colors $C_{1..3}$ for increasing values of α. The row *Success* indicates the probability of completing a tunnel without hitting obstacles

	$\alpha=0$	$\alpha=0.1$	$\alpha=0.5$	$\alpha=1$	$\alpha=2$	$\alpha=5$
C_0	0.8	0.8	0.76	0.76	0.77	0.75
$C_{1..3}$	0.29	0.21	0.15	0.14	0.13	0.11
Success	0.99	0.95	0.96	0.95	0.92	0.87

Table 3 Policy legibity for increasing values of α

	$\alpha = 0$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$	$\alpha = 5$
$\mathcal{L}$	0.30	0.36	0.44	0.48	0.48	0.51
$\mathcal{L}_{\text{gain}}$	1	1.2	1.46	1.6	1.6	1.7

possible in advance to a certain degree, because the agent starts avoiding cells before actually walking on them but as soon as they enter its field of view.

In addition, to better understand the effect of regularization, the agent's behaviors for three different configurations of colored regions and increasing factor α are plotted in Fig. 7. The plots have two colors: red as reward color for the agent's policy, and blue as reward color for a different policy. Legibility clearly skews the trajectories such that they pass farther away from non-red cells in a way that is proportional to α. However, regularization becomes detrimental for values of α that are too high. In such cases, the agent's original policy of walking over red cell is dominated by the regularization to avoid blue cells, and in some cases the agent is not able to pass over any red cells even if there aren't any obstacles.

Notice how this type of reward regions simulates goal locations, and thus allows to qualitatively confront the here obtained legible behavior with those for goal-driven policies from literature [27, 28]. The behaviors are quite similar, with trajectories that are arced to disambiguate the goals. Crucially, previous methods require to retrain the agent each time the goals or environment change, because the regularization is embedded in the training procedure of the agent. In our method this is not necessary.

5 Discussion

Our quantitative results indicate that the proposed regularization for legibility is effective in making the observer model discriminate the true agent's policy. This is highlighted in Table 2 where we can see that the reward ratio for colors different from C_0 decrease as α increases, signifying that the agent avoids regions with colors different from its own. The qualitative results also confirm this observation, by showing that as α increases so does the effort of the regularized policy to avoid other colors.

We calculated how high values of α are detrimental both in terms of accumulated reward and success rate of the episodes. The reason is that the agent is regularized so much to avoid other policies that its original policy is overridden rather than regularized. However, the agent incurs a noticeable loss in terms of accumulated reward or success rate only for high regularization factors.

The general results confirm our originally formulated hypothesis that legibility increases the probability of the agent policy appearing in the observer's inferences by making the agent avoid rewarding regions of other policies. This is an emergent behavior that was not coded in the equations and in our experiments represented a generalization over goal-driven solutions for legibility. This is because the reward regions of goal-driven policies, i.e., having reward regions at goal locations, are a special case of those of arbitrary policies, that can instead have reward regions anywhere in the environment.

Furthermore, we noticed that our results are obtained with a simple observer model that considers the same policies that are available to the agent but with a uniform probability. Because our results are qualitative similar to

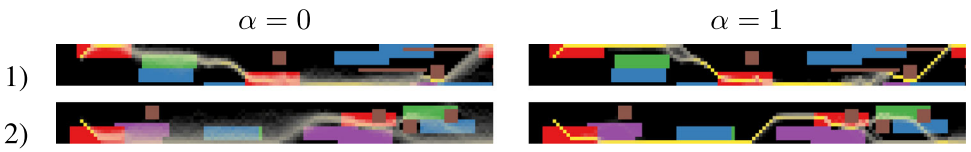

Fig. 6 Qualitative results for two sampled tunnels. The left column shows the agent's optimal learned behavior while the right column its regularized version

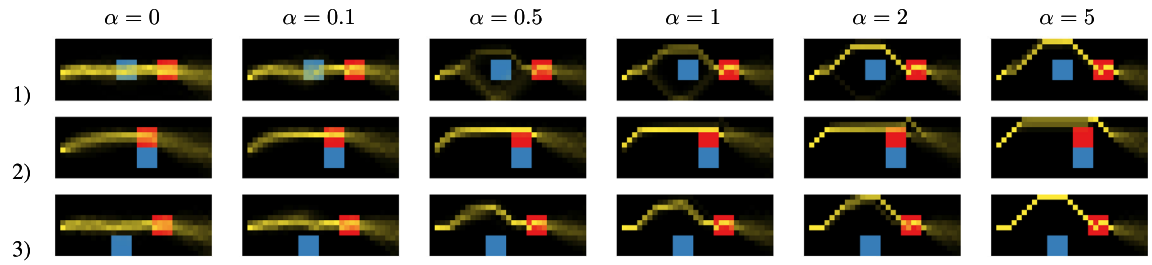

Fig. 7 Qualitative results for three types of positions of reward regions and increasing levels of α

those reported in earlier works on legibility [11, 27, 28] i.e., legible trajectories are skewed to avoid other goal locations, it is suggested that a similarly uniformly initialized observer model was implicitly utilized in those papers as well. However, we regularize the agent by working on policies rather than reward distributions, and for this reason the proposed method has three major advantages: firstly it easily generalizes over different shapes of reward regions and not only only on goal states that are a particular type of reward region. Then, it allows to easily generalize over differences between agent and observer models, that is by using different corresponding Q-networks. This was not possible in previous methods because the observer was imprinted in the agent's Q-values during training. And finally, it allows to regularize the agent for arbitrary positions of rewards and obstacles without requiring to retrain the agent for each of them.

6 Conclusion

In this paper we introduced a model that allows to regularize a reinforcement learning agent for legibility. In our formulation we propose a legibility criteria that induces an observer model to disambiguate the agent's intention from a set of others, with intentions being implemented as policies. We suggest that rather than modifying the learning procedure of the agent we can wrap a priorly learned set of policies by a pair of Bayesian Networks that model agent and observer, respectively. The coupled networks describes a setting of second-order theory of mind that, by reasoning on how the observer infers policies, increases the discrimination between the agent's true policy and other candidate policies.

We evaluated the method on an illustrative example showing how legibility impacts the decision boundary of the agent, and on a Deep-RL example. In general, our model is successful at increasing the legibility of trajectories without incurring in losses for the agent when the regularization factor is kept at a reasonable level. Furthermore, our qualitative results show that the obtained trajectories are similarly arced as those obtained in earlier work on Explainable Planning, but with the main difference of computing legibility on reward regions rather goal states.

The proposed methods introduces three relevant degrees of freedom in legibility. The first is that legibility is computed with respect to reward regions rather than goal locations. This allows to regularize arbitrary policies and especially those that can run indefinitely. Policies of this type can't be regularized by methods relying on the original formulation of legibility because of the need of a goal state. The second degree of freedom is on the possibility of decoupling agent and observer models. This allows to specify that the the observer uses a different reward distribution, and legibility is to be computed against that distribution rather than the agent's. This decoupling is not easy to implement using previous methods relying on distance measures computed on the Cartesian space, because would require to specify how the observer measures distances on the state-space differently from the agent. Finally, since we don't rely on augmenting the agent's reward distribution for legibility, but regularize its policy after training, our method results applicable on all combinations of the environment without retraining the agent. Since the agent's learning algorithm is unmodified, it is straightforward to apply our method to arbitrary problems and types of agents. Even though we couldn't test it on extensive test beds of agents and problems, it is reasonable to think that problems effectively captured as MDPs can be regularized without major additional implementations.

7 Future work

Possible future work relates to test legibility on settings striving toward real scenarios. In particular, it is relevant to test highly nonlinear environments, such as those from Atari games. In these environments the agent emergently develops behaviors that are more symbolic rather than purely reactive. For example, it may learn to focus on a particular region of the environment (thus creating a subgoal, *Breakout*), or learn to perform sequences of actions achieving sub-goals one at a time (*Montezuma Revenge*), [34, 35]. How would legibility behave in such scenarios? We could expect a similar behavior, avoiding to communicate wrong policies (or plans). However, due to the increased complexity, this could require to leverage observer models that explains the agent symbolically e.g., through planning, rather than as an MDP. This is possible in the mirror setting by changing the observer model into one that leverages symbolic reasoning, yet maintaining compatibility between nodes to compute the divergences. In alternative, we can think of learning, in the mirror setting, symbolic observer models best explaining the agent's behavior, thus extracting the planning model embedded in the agent's neural network.

Another interesting line of research reformulates legibility as control mechanism for the agent, allowing to dynamically regularize its behavior in real-time. This is possible by allowing a controller to explicitly set part of the observer's network, thus constraining the regularization process towards a desired outcome (e.g., perform while avoiding blue cells, or perform while walking on red cells). In general we see implications for research on the gap

between symbolic and sub-symbolic reasoning and for supporting explainability in RL.

Funding Open access funding provided by Umea University. This study was funded by the Department of Computing Science of Umeå University, Universitetstorget 4, Umeå, 901 87, Sweden.

Declarations

Conflict of interest The author declares that there are no conflicts of interest associated with this study.

Human participants or animals This study did not involve human participants or animals.

References

1. Anjomshoae S, Najjar A, Calvaresi D, Främling K (2019) Explainable agents and robots: Results from a systematic literature review. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. AAMAS '19, pp. 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC
2. Hellström T, Bensch S (2018) Understandable robots-what, why, and how. Paladyn J Behav Robot 9(1):110–123
3. Schaefer KE, Straub ER, Chen JY, Putney J, Evans AW III (2017) Communicating intent to develop shared situation awareness and engender trust in human-agent teams. Cognit Syst Res 46:26–39
4. Chang ML, Gutierrez RA, Khante P, Short ES, Thomaz AL (2018) Effects of integrated intent recognition and communication on human-robot collaboration. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3381–3386. IEEE
5. Sciutti A, Mara M, Tagliasco V, Sandini G (2018) Humanizing human-robot interaction: on the importance of mutual understanding. IEEE Technol Soc Mag. 37(1):22–29
6. Losey DP, McDonald CG, Battaglia E, O'Malley MK (2018) A review of intent detection, arbitration, and communication aspects of shared control for physical human-robot interaction. Appl Mech Reviews 70(1)
7. Verberne FM, Ham J, Midden CJ (2012) Trust in smart systems: sharing driving goals and giving information to increase trustworthiness and acceptability of smart systems in cars. Human Factors 54(5):799–810
8. Walker M, Hedayati H, Lee J, Szafir D (2018) Communicating robot motion intent with augmented reality. In: Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, pp. 316–324
9. Kulkarni A, Zha Y, Chakraborti T, Vadlamudi SG, Zhang Y, Kambhampati S (2019) Explicable planning as minimizing distance from expected behavior. In: AAMAS, pp. 2075–2077
10. Zhang Y, Sreedharan S, Kulkarni A, Chakraborti T, Zhuo HH, Kambhampati S (2017) Plan explicability and predictability for robot task planning. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 1313–1320. IEEE
11. Dragan AD, Lee KC, Srinivasa SS (2013) Legibility and predictability of robot motion. In: 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 301–308. IEEE
12. Persiani M, Hellström T (2021) Probabilistic plan legibility with off-the-shelf planners. In: 9th ICAPS Workshop on Planning and Robotics. ICAPS 2021
13. Puiutta E, Veith EM (2020) Explainable reinforcement learning: A survey. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction, pp. 77–95. Springer
14. Cashmore M, Collins A, Krarup B, Krivic S, Magazzeni D, Smith D (2019) Towards explainable ai planning as a service. arXiv preprint arXiv:1908.05059
15. Alharin A, Doan T-N, Sartipi M (2020) Reinforcement learning interpretation methods: a survey. IEEE Access 8:171058–171077
16. Du M, Liu N, Hu X (2019) Techniques for interpretable machine learning. Commun ACM 63(1):68–77
17. Roscher R, Bohn B, Duarte MF, Garcke J (2020) Explainable machine learning for scientific insights and discoveries. IEEE Access 8:42200–42216
18. Mott A, Zoran D, Chrzanowski M, Wierstra D, Jimenez Rezende D (2019) Towards interpretable reinforcement learning using attention augmented agents. Adv Neural Inf Proces Syst 32:12350–12359
19. Liu G, Schulte O, Zhu W, Li Q (2018) Toward interpretable deep reinforcement learning with linear model u-trees. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 414–429. Springer
20. Stiglic G, Kocbek P, Fijacko N, Zitnik M, Verbert K, Cilar L (2020) Interpretability of machine learning-based prediction models in healthcare. Wiley Interdiscip Rev: Data Min Knowl Discov 10(5):1379
21. Rutherford MD (2004) The effect of social role on theory of mind reasoning. Br J Psychol 95(1):91–103
22. Perez-Osorio J, Wykowska A (2020) Adopting the intentional stance toward natural and artificial agents. Philos Psychol 33(3):369–395
23. Firestone C (2020) Performance vs. competence in human-machine comparisons. Proc National Acad Sci 117(43):26562–26571
24. Chakraborti T, Kulkarni A, Sreedharan S, Smith DE, Kambhampati S (2019) Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior. In: Proceedings of the International Conference on Automated Planning and Scheduling, vol. 29, pp. 86–96
25. MacNally AM, Lipovetzky N, Ramirez M, Pearce AR (2018) Action selection for transparent planning. In: AAMAS, pp. 1327–1335
26. Chakraborti T, Sreedharan S, Zhang Y, Kambhampati S (2017) Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In: 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, pp. 156–163. International Joint Conferences on Artificial Intelligence
27. Bied M, Chetouani M (2020) Integrating an observer in interactive reinforcement learning to learn legible trajectories. In: 2020

29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pp. 760–767. IEEE
28. Zhao X, Fan T, Wang D, Hu Z, Han T, Pan J (2020) An actor-critic approach for legible robot motion planner. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 5949–5955. IEEE
29. Kostrikov I, Fergus R, Tompson J, Nachum O (2021) Offline reinforcement learning with fisher divergence critic regularization. In: International Conference on Machine Learning, pp. 5774–5783. PMLR
30. Wu Y, Tucker G, Nachum O (2019) Behavior regularized offline reinforcement learning. arXiv preprint arXiv:1911.11361
31. Mysore S, Mabsout B, Mancuso R, Saenko K (2021) Regularizing action policies for smooth control with reinforcement learning. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 1810–1816. IEEE
32. Szepesvári C (2010) Algorithms for reinforcement learning. Synth Lect Artif Intell Mach Learn 4(1):1–103
33. Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W (2016) Openai gym. arXiv preprint arXiv:1606.01540
34. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602
35. Aytar Y, Pfaff T, Budden D, Paine T, Wang Z, De Freitas N (2018) Playing hard exploration games by watching youtube. Adv Neural inf Process Syst, 31