Interestingness Elements for Explainable Reinforcement Learning: Understanding Agents' Capabilities and Limitations

Pedro Sequeira and Melinda Gervasio
SRI International
333 Ravenswood Avenue, Menlo Park, CA 94025, United States
pedro.sequeira@sri.com, melinda.gervasio@sri.com

December 20, 2019

Abstract

We propose an explainable reinforcement learning (XRL) framework that analyzes an agent's history of interaction with the environment to extract interestingness elements that help explain its behavior. The framework relies on data readily available from standard RL algorithms, augmented with data that can easily be collected by the agent while learning. We describe how to create visual explanations of an agent's behavior in the form of short video-clips highlighting key interaction moments, based on the proposed elements. We also report on a user study where we evaluated the ability of humans in correctly perceiving the aptitude of agents with different characteristics, including their capabilities and limitations, given explanations automatically generated by our framework. The results show that the diversity of aspects captured by the different interestingness elements is crucial to help humans correctly identify the agents' aptitude in the task, and determine when they might need adjustments to improve their performance.

1 Introduction

Reinforcement learning (RL) is a popular computational approach for autonomous agents facing a sequential decision problem in dynamic and often uncertain environments [31]. The goal of any RL algorithm is to learn a *policy*, *i.e.*, a mapping from states to actions, given trial-and-error interactions between the agent and an environment. Typical approaches to RL focus on memoryless (reactive) agents that select their actions based solely on their current observation [22]. This means that by the end of learning, an RL agent can select

1

the most appropriate action in each situation—the learned policy ensures that doing so will maximize the reward received by the agent during its lifespan, thereby performing according to the underlying task assigned by its designer.

RL agents do not need to plan or reason about their future to select actions,[1] which makes it hard for them to explain their behavior—all they know is that they *should* perform a particular action in a situation, in the case of deterministic policies, or select an action according to a probability distribution, in the case of stochastic policies. The "why" behind decision-making is lost during the learning process as the policy *converges* to an optimal action-selection mechanism. At most, agents know that choosing one action is preferable over others, or that some actions are associated with a higher value—but not why that is so or how it came to be. RL further complicates explainability by enabling an agent to learn from *delayed rewards* [31]—the reward received after executing some action is "propagated" back to the states and actions that led to that situation, meaning that important actions may be associated with any (positive) reward.

Ultimately, RL agents lack the ability to know why some actions are preferable over others, to identify the goals that they are currently pursuing, to recognize what elements are more desirable, to identify situations that are "hard to learn", or even to summarize the strategy learned to solve the task.

This lack of self-explainability can be detrimental to establishing trust with human collaborators who may need to delegate critical tasks to agents. An eXplainable RL (XRL) system that enables humans to correctly understand the agent's *aptitude* in a specific task, *i.e.*, both its capabilities and limitations, whether innate or learned, will let them delegate tasks more appropriately as well as identify situations where the agent's perceptual, actuating, or control mechanisms, may need to be adjusted prior to deployment.

1.1 Approach Overview

In this paper, we present a framework towards making autonomous RL agents more explainable through introspective analysis (Fig. 1).[2] The agent examines its history of interaction with the environment to extract *interestingness elements*[3] denoting meaningful situations, *i.e.*, potentially "interesting" characteristics of the interaction. Each element is generated by applying statistical analysis and machine learning methods to data collected *during* the interaction. As depicted, the introspection framework is the basis of an explanation system operating over the interestingness elements, selecting the ones that best explain the agent's behavior.

---

[1]While model-based RL methods such as prioritized sweeping [25] do use planning to update a value function, here we mean the classical notion of planning where agents plan or reason about a sequence of actions to be executed. In this sense, *after learning*, RL agents simply rely on the policy to decide which action to take at each step, hence they do not need to plan.

[2]In this paper we detail only the elements evaluated in our user study. We refer to A for additional elements of different types and to [30] for a preliminary version.

[3]We borrow the term from the association rule mining literature [see *e.g.*, 4] to denote potentially relevant aspects of the agent's history of interaction with the environment.
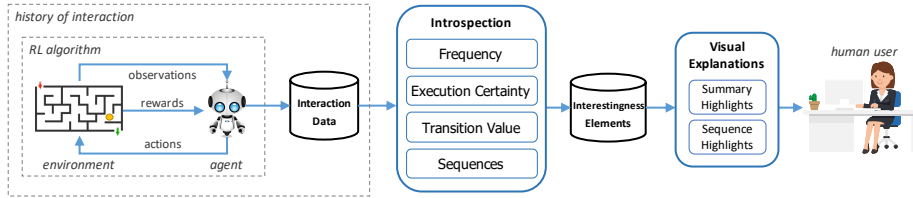
Figure 1: Our framework for explainable autonomous RL agents. During its interaction with the environment, the agent collects relevant statistical information. Our introspection framework analyzes the collected data and identifies interestingness elements of the history of interaction. These elements are then used to explain an agent's learned behavior and its capabilities and limitations in the task to a human user.

To assess the usefulness of the proposed framework, we conducted a user study to evaluate visual explanations, in the form of short video clips, automatically generated by our XRL framework. First, we designed agents with different goals and perceptual characteristics in a video game task. The goal was to simulate players with different capabilities and limitations, and with different needs for improvement in each of the game's sub-tasks. We trained each agent using standard RL, resulting in distinct behaviors and levels of game performance. We then conducted an online survey where we asked subjects to observe several videos, each depicting the behavior of a particular agent and generated based on specific interestingness elements.

Overall, our results show that different agent characteristics require different elements to be presented to a user to enable a correct understanding of proficiency in different parts of a task. Presenting only "good" or only "bad" instances of agent behavior can lead users to form incorrect expectations about agent performance. The results also showed that the frequency of observations can help select diverse moments of an agent's interaction with the environment, allowing for a correct understanding of its general aptitude. Moreover, short summaries showing how an agent overcomes challenges and achieves its goals can provide a short but accurate depiction of its learned strategy.

The paper is organized as follows. Sec. 2 introduces the necessary concepts behind the RL framework. Sec. 3 details the proposed interestingness elements while Sec. 4 shows how visual explanations can be derived from them. Sec. 5 presents the experimental study, including the design of the different RL agents and the survey. Sec. 6 details the results of the user study and provides an in-depth discussion of the main insights stemming from the results. Sec. 7 discusses related works and compares them with our approach. Finally, Sec. 8 summarizes the main findings and describes current and future developments.

3

2   Reinforcement Learning

We now introduce the necessary notation and terminology used throughout the paper related to the computational framework of RL. RL agents can be modeled using the *Markov decision process* (MDP) framework [27]. Formally, we denote an MDP as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathsf{P}, \mathsf{R}, \gamma)$, where: $\mathcal{S}$ is the set of all possible environment states; $\mathcal{A}$ is the action repertoire of the agent; $\mathsf{P}(s' \mid s, a)$ indicates the probability that the state at time step $t + 1$, S_{t+1}, is s', given that the state at time step t, S_t, is s and that the agent selected action $A_t = a$ at t. $\mathsf{R}(s, a)$ represents the average reward that the agent expects to receive for performing action a in state s; $0 \leq \gamma < 1$ is some discount factor denoting the importance of future rewards.

An MDP evolves as follows. At each time step $t = 0, 1, 2, \ldots$, the environment is in some state $S_t = s$. The agent selects some action $A_t = a \in \mathcal{A}$ and the environment transitions to state $S_{t+1} = s'$ with probability $\mathsf{P}(s' \mid s, a)$. The agent receives a reward $\mathsf{R}(s, a) = r \in \mathbb{R}$, and the process repeats.

In this paper, we deal with the problem of explainability in situations where the agent may have limited sensing capabilities that prevent it of completely determining the current state of the system, *i.e.*, the environment may be *partially-observable*. Without loss of generality, we assume that the agent's observations are sufficient for it to solve the intended task, in which case they are treated *as if they were equal to* the underlying true states. We use $\mathcal{Z} = \mathcal{S}$ to denote the set of all possible observations and symbol z to refer to singular observations.

The goal of the agent can be formalized as that of gathering as much reward as possible throughout its lifespan discounted by γ. This corresponds to maximizing the value $v = \mathbb{E}\left[\sum_t \gamma^t r\right]$, where the reward r evaluates the immediate utility of making action a in state s at time-step t, in light of the underlying task that the agent must solve. In order to maximize v, the agent must learn a policy, denoted by $\pi : \mathcal{Z} \rightarrow \mathcal{A}$, that maps each observation $z \in \mathcal{Z}$ directly to an action $\pi(z) \in \mathcal{A}$. In the case of MDPs, this corresponds to learning a policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ referred to as the *optimal policy* maximizing the value v.

In this paper, we focus on *value-based* RL methods where there is a function $Q^* : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ associated with π^* that verifies the recursive relation:

$$Q^*(s, a) = r + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s' \mid s, a) \max_{b \in \mathcal{A}} Q^*(s', b). \tag{1}$$

$Q^*(s, a)$ represents the *value* of executing action a in state s and henceforth following the optimal policy. RL algorithms like Q-learning [35] assume that the agent has no knowledge of either P or R. Hence, they typically start by *exploring* the environment—selecting actions in some exploratory manner—collecting samples in the form (s, a, r, s') which are then used to successively approximate Q^* using Eq. 1. After exploring its environment, the agent can *exploit* its knowledge and select the actions that maximize (its estimate of) Q^*.

3 Introspection Framework

In this section we describe our introspection framework, including the generated interestingness elements.

3.1 Interaction Data

As shown in Fig. 1, the introspection framework relies on the following data collected by the agent during its interaction with the environment:

- $n(z)$: the number of times the agent observed z; $n(z,a)$: the number of times it executed action a after observing z; and $n(z,a,z')$: the number of times it observed z' after executing action a when observing z;

- $\hat{\mathsf{P}}(z' \mid z,a)$: the *estimated probability* of observing z' when executing action a after observing z. This can be modeled from the interaction according to $\hat{\mathsf{P}}(z' \mid z,a) = n(z,a,z')/n(z,a)$;

- $Q(z,a)$: the agent's *estimate of the Q function*, corresponding to the expected value of executing a having observed z and henceforth following the current policy. This can be estimated using any value-based RL algorithm;

- $V(z)$: the agent's *estimate of the V function* that indicates the value of observing z and henceforth following the current policy. This corresponds to $V(z) = \max_{a \in \mathcal{A}} Q(z,a)$;

As we can see, some of the interaction data is already collected by value-based RL methods, namely the Q function, and by model-based algorithms, namely $\hat{\mathsf{P}}$ and $\hat{\mathsf{R}}$. The remaining data can easily be collected by the agent during its interaction with the environment by updating counters and running averages.

As shown in Fig. 1, all this data is used by the agent for various introspective analyses. The idea is to extract the most out of the statistical information collected during the interaction to highlight relevant information that might help explain the agent's behavior.

Table 12 lists the analyses considered in this paper and a short description of the elements that each generates. In the continuation, we present a mathematical interpretation for the several interestingness elements and discuss how each can be used to help explain the agent's interaction with an environment.

3.2 Frequency Analysis

(In)Frequent situations. These correspond to observations that appeared less/ more frequently than others during the interaction. To help explain the agent's behavior based on this dimension one can select these elements according to whether the frequency in the tables $n(z)$, $n(z,a)$ and $n(z,a,z')$ is below/above a given threshold, or select the top/bottom k observations.

Table 1: Overview of the dimensions of analysis evaluated in the user study and the generated interestingness elements.

Dimension	Generated Interestingness Elements
Frequency	(in)frequent situations: *situations the agent finds very (un)common*
Execution Certainty	(un)certain executions: *hard/easy to predict with regards to action execution*
Transition-Value	observation minima and maxima: *learned goals and situations to avoid*
Sequence	most likely sequences to maxima: *summaries of the agent's learned strategies*

Explainability purpose: this element can be used to expose the agent's (in)experience with its environment, denoting both its commonly and rarely encountered situations. The latter may indicate states that were not sufficiently explored by the agent, *e.g.*, locations that are hard to reach in a maze. It can also identify situations that had such a negative impact on the agent's performance (*e.g.*, a death situation in a game) that its action-selection and learning mechanisms made sure they were rarely visited.

It can also be used during learning to denote novel or rare situations, which may indicate that the environment is highly dynamic or that the agent is exploring. In an interactive setting, these elements provide opportunities for an agent to ask a user for guidance, *e.g.*, in choosing the appropriate actions in infrequently encountered situations.

3.3 Execution Certainty Analysis

(Un)Certain executions. These estimate the (un)certainty of each observation with respect to action execution. Observations where many different actions are executed often (high dispersion) are considered uncertain, while those in which only a few actions are selected are considered certain. Formally, given an observation z, the execution certainty associated with z is measured by the concentration of the executions of the actions $a \in \mathcal{A}$. We use the *evenness* of the distribution over next observations as given by its normalized true diversity (information entropy) [26]. Formally, let $p(X)$ be a probability distribution over $x_i \in X, i = 1, \ldots, N$ of set X. The evenness of p over X is then provided by:

$$\xi(X) = -\sum_{x_i \in X^+} p(x_i) \ln p(x_i) / \ln N, \tag{2}$$

where $X^+ \doteq \forall_{x_i \in X} \colon p(x_i) > 0$. We then use this evenness measure to calculate the dispersion of the distribution over actions according to $\xi_z = \xi(\pi(z))$, where π is any policy of interest. In our study we approximate the agent's interaction

policy using $\hat{\pi}(z) = n(z, \mathcal{A})/n(z)$. We argue that this formulation retains information about the agent's history of interaction beyond the learned "optimal" policy by, *e.g.*, capturing situations that were harder to learn.

Explainability purpose: this element can be used to reveal the agent's confidence in its decisions. Situations where the agent is uncertain of what to do indicate opportunities to ask a user for help. They are also particularly important because people tend to require explanations mostly for abnormal behavior [24]. On the other hand, certain situations correspond to what the agent has learned well and where its behavior is more predictable.

3.4 Transition-Value Analysis

Observation minima and maxima. These refer to observations whose values are lower/greater than or equal to the values of all possible next observations, as informed by the agent's learned predictive model. Formally, let $\mathcal{T}_z \doteq \{\forall_{z' \in \mathcal{Z}} : \exists_{a \in \mathcal{A}} \hat{P}(z' \mid z, a) > 0\}$ be the set of observed transitions starting from observation z. The *local minima* are defined by $\mathcal{Z}_{min} \doteq \forall_{z \in \mathcal{Z}} : \forall_{z' \in \mathcal{T}_z} V(z) \leq V(z')$. Similarly, the *local maxima* are defined by $\mathcal{Z}_{max} \doteq \forall_{z \in \mathcal{Z}} : \forall_{z' \in \mathcal{T}_z} V(z) \geq V(z')$.

Explainability purpose: these elements may help explain the desirability attributed by the agent to a given situation. Specifically, observation maxima denote sub-goals or acquired preferences for the agent, identifying situations where the agent prefers to remain in the same state rather than explore the surrounding environment. Exposing such situations to a user enables observing the agent performing at its best according to what it has learned. This lets users verify whether performance is in line with expectations, or that adjustment of the agent's training, reward function, perceptual characteristics, etc. is needed.

Observation minima denote highly undesirable situations that the agent wants to avoid and where taking any action leading to a different state is preferable. By observing the agent's behavior in such situations, a user can understand how well the agent handles difficult situations, and whether further adjustments are required.

3.5 Sequence Analysis

The goal of this analysis is to calculate common and relevant sequences of actions learned by the agent. In particular, these sequences involve starting from important observations identified by the other analyses, and then performing actions until a local maximum, *i.e.*, a *goal state*, is reached. We first create a state-transition graph where nodes are observations $z \in \mathcal{Z}$ and edges are actions $a \in \mathcal{A}$ denoting the observed transitions, weighted according to the probability $\hat{P}(z' \mid z, a)$. The resulting graph is directed with nonnegative weights, making it amenable to best-first search algorithms to find the most likely sequence of observation-action pairs between two given observations.

We use a variant of Dijkstra's algorithm [8], whose input is a source observation $z_s \in \mathcal{Z}_s$, where $\mathcal{Z}_s$ is an observation set of interest, and a set of possi-

ble target observations, $\mathcal{Z}_{max}$, corresponding to the maxima discovered by the transition-value analysis. First, we determine the most likely paths between z_s and each target observation $z_t \in \mathcal{Z}_{max}$. Let $P_{st} = [z_0 = z_s, a_1, z_1, \ldots, a_n, z_n = z_t]$ denote a path between z_s and z_t. The probability of observing z_t after observing z_s and following path P_{st} is thus given by $p(z_s, z_t) = \sum_{i=1}^{|P_{st}|} \hat{\mathsf{P}}(z_i \mid z_{i-1}, a_i)$. We then choose the *most likely path weighted by value*, connecting the source and an optimal target observation, according to: $z_t^* = \mathrm{argmax}_{z_t \in \mathcal{Z}_{max}} p(z_s, z_t) V(z_t)$. Using this procedure, we generate the following element:

Most likely sequences to maxima. These denote sequences starting from an observation of interest, and then performing actions until a subgoal is reached. The source observation set can include the (in)frequent, (un)certain, or minima observations discovered by the other analyses. Sequences thus denote the agent's typical or most likely course of action from particular situations.

Explainability purpose: this element can be used to reveal more purposeful agent behavior to a human user, *e.g.*, performing actions from a low-valued situation toward a goal. This avoids requiring the user to view complete episodes while providing potentially more information for understanding the agent's behavior compared to viewing performance in the context of a single element.

The sequence-finding procedure can also be used by the user to query the agent about its future goals and behavior in *any* possible situation. Notably, this can be used to provide contrastive explanations which help reason about why the alternatives to some actions—the foils—are not as desirable as those chosen by the agent [24]. The starting points may denote *reasons* for behavior while the combined transition likelihoods denote the agent's *beliefs*—two crucial elements used by people to explain intentional events [7].

4 Visual Explanations

As depicted in Fig. 1, provided that the behavior of the agent can be readily visualized, the interestingness elements extracted by our analyses can be used to generate visual explanations. This type of explanation is therefore suitable for agents relying on a visual input such as robots or game-playing agents. We follow an approach similar to [3], where short video-clips highlighting the behavior of the agent during key moments of interaction are recorded.

The rationale behind this approach is that simply watching the behavior of the agent performing the task, *e.g.*, during whole episodes, might be exhausting for a user and mostly uninformative. Therefore, the system is given a *budget* for a given explanation. The budget will typically be application-dependent and can be defined in terms of, *e.g.*, the maximum length of the video clip or the maximum number of time-steps or highlights per explanation. Two types of video highlights can be generated by our framework: summaries and sequences.

4.1   Summary Highlights

These highlights combine different examples of the agent's behavior. Like the approach in [3], given some history of interaction with an environment, *e.g.*, a sequence of simulated episodes, an explanation is a summary composed of k highlights (agent trajectories) of length l time-steps, where $(l-1)/2$ time-steps occur before and after the situation of interest. The highlights are then combined into a single video clip for visualization.

Departing from [3], we introduce three novel ways in which the key moments are selected. First, as mentioned earlier, each interestingness element is defined according to some analysis dimension. For example, frequency elements can be sorted from the most to the least frequent observations. During highlight selection, we try to sample situations closer to the extrema—*e.g.*, when highlighting frequent observations, a trajectory passing through the most frequent observation is the most preferred. The idea is to select the most representative examples of each source of interestingness.

Second, for explanations where $k > 1$, we use a secondary selection criterion of *diversity*. Namely, whenever a situation for a particular interestingness element is experienced and the maximum number of k highlights for that element type has been reached, the explanation decides whether to replace any of the existing highlights with the new observation. Formally, let $d(z_1, z_2) \in [0, 1]$ be an observation diversity/distance metric relevant to the task. Then, given the $k + 1$ highlights, we keep the set of k highlights maximizing:

$$\max_{z_i, z_j} d(z_i, z_j) \times \min_{z_i, z_j} d(z_i, z_j), \quad i \neq j, \quad i, j = 0, \dots, k - 1$$

The idea is to maximize the total diversity of the final set of highlights such that the user can observe different nuances of the agent's learned behavior.

Finally, we introduce the step of deciding how to visualize and compile the highlights. When experimenting with visualization, we noticed that, depending on the task, the user sometimes fails to understand which part of the agent's behavior is being highlighted. This can be critical in highly dynamic environments, where the behavior of the agent can vary dramatically at each time-step. To mitigate this problem, we introduce fade-in/fade-out effects around each highlight so that the user can focus on the important moment being highlighted while also being presented with the context in which it occurred.

4.2   Sequence Highlights

Our framework provides an additional type of highlighting capability based on the element of likely sequences to goal, which enables highlighting relevant sequences of behavior rather than single moments. In particular, in our user study we focus on sequences starting in an observation *minimum* and ending in a *maximum*, as informed by the transition-value analysis. This reveals examples of the agent's learned strategy, *i.e.*, a trajectory showcasing its capabilities in overcoming perceived difficulties and in reaching its learned goals.

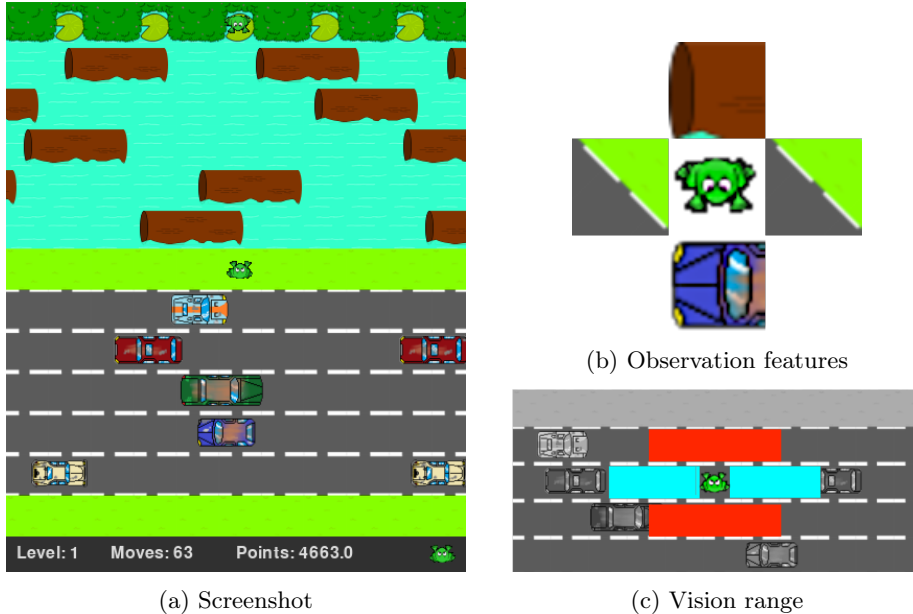(a) Screenshot

(b) Observation features

(c) Vision range

Figure 2: The Frogger game task (left) and agent design aspects (right).

5 Experimental Study

To evaluate the usefulness of our XRL framework in helping humans correctly understand the aptitude of different RL agents, we conducted a user study using a simple video-game scenario based on Frogger.[4] Fig. 2a shows a screenshot of the game. The player is responsible for controlling frogs, one at a time, to go from the grassy strip at the bottom to the lilypads at the top, with only one frog at a time allowed on a lilypad. The player has to first cross the road without getting hit by cars and then cross the river by jumping on logs. The control set is $\mathcal{A} = \{N, S, E, W\}$, where each action deterministically moves the frog by 40 pixels in the corresponding direction.

When a frog reaches a lilypad, a new frog appears at the bottom of the screen. When two frogs reach the lilypads, the agent completes the current level and starts the next level. This clears the lilypads and makes cars and logs go faster. A player has a limit of 100 moves to pass a level and 3 lives in a game. The player loses a life when the maximum number of moves is reached, a frog is hit by a car, a frog jumps into the river, or a frog is on a log that goes off-screen. At the beginning of each game, the time interval at which logs are introduced in the environment is randomized.

---

[4]We used the implementation in https://github.com/pedrodbs/frogger.

5.1 Agent Experiments

To make the Frogger game an interesting task from the perspective of behavior explanation, we designed agents with *partial observability*. Specifically, each observation $z \in \mathcal{Z}$ provides a local view of the environment according to the frog's location. Observations are composed of four discrete features, denoted by $z = [\phi_N, \phi_S, \phi_E, \phi_W]$, each $\phi_i \in \Phi$ indicating the element that is visible in the corresponding direction, where $\Phi = \{empty, water, car, log, lilypad, bounds\}$ is the feature set, $bounds$ means that the agent is near the environment's borders, and $empty$ that no element is visible in that direction.[5]

This partial observability introduces uncertainty in the environment's dynamics *from the agent's perspective*. Fig. 2b shows what an agent observes given the true game state in Fig. 2a. As we can see, $\phi_S = car$ although the car is not directly beneath the frog. This has to do with the way we designed the perception of cars, where two parameters control for an agent's vision range, as illustrated in Fig. 2c. The blue squares indicate the agent's *horizontal vision range*, which is defined by parameter vis_h. This is the horizontal distance, in pixels, below which an agent is capable of seeing cars in its E and W directions, *i.e.*, a car in the same lane. The red squares indicate the *vertical vision range*, which is defined by parameter vis_v. This determines the horizontal distance, in pixels, below which an agent considers that a car is present in its N and S directions, *i.e.*, in the lane *directly* above and below the agent, respectively. As for the logs, we designed the log feature such that an agent anticipates the movement of adjacent logs.[6]

The reward function R was defined as follows. The agent receives a reward $r(z, a) = -200$ in situations where executing a when observing z results in one of the aforementioned death conditions. A punishment of -300 occurs when the agent depletes its available lives and a reward of $5,000$ is received whenever a frog arrives on a lilypad. A reward of -1 is also received at each time-step.

5.1.1 Agent Types

To study the impact of the different interestingness elements in helping understand the aptitude of RL agents, we designed three different agents:

Optimized: this agent can observe cars at a moderate distance, *i.e.*, when they are neither too close nor too far away. Its parameters were empirically fine-tuned to achieve a very high performance in the task. The agent uses the default reward function.

High-vision: this agent has very high vision capability, *i.e.*, it can anticipate the presence of cars from afar. It was designed to simulate an agent with

---

[5]The agent does not distinguish between the road and grass for feature $empty$, nor the type of car for the car feature.

[6]The agent's observations are extracted without access to the game engine's internal dynamics, including the logs' velocities, hence they are not perfect. This means there is still a chance ($p > 0$) that performing action N when $\phi_N = log$ will result in a death in the river.

Table 2: The parameterization of each agent in our experiments. Legend: r_{river}: reward received after a death in the river; q_{init}: Q-values initialization. See text for more details.

Agent	vis_h	vis_v	r_{river}	q_{init}
Optimized	60	40	-200	$5,000$
High-vision	140	120	-200	$5,000$
Fear-water	60	40	$-10,000$	0

inadequate perceptual capabilities, *i.e.*, whose sensing mechanisms were poorly calibrated to the task. It also uses the default rewards.

Fear-water: this agent has the same visual capabilities as the optimized agent. It also uses the default reward function with one exception: when it dies on the river, it receives a punishment of $r = -10,000$. This simulates a motivational impairment—the rewards idealized by its designer are not appropriate for the agent to learn the intended task.

A detailed parameterization of each agent is listed in Table 2.

5.1.2   Agent RL Training

Each agent was trained using standard Q-learning [35]. To promote exploration, we used optimistic initialization by setting $Q(z, a) = q_{init}$ for all observations $z \in \mathcal{Z}$ and actions $a \in \mathcal{A}$, where q_{init} controls the agent's optimism. Each agent was first trained for $2,000$ episodes, with each episode corresponding to one game, *i.e.*, ending when the agent lost all of its 3 lives or when a limit of 300 time-steps was reached. We used a Softmax action-selection mechanism with exponentially-decaying temperature β, *i.e.*, given training episode e, the temperature is given by $\beta = \beta_{\min} + \beta_{\max}0.995^e$. In our experiments, we set $\beta_{\min} = 0.05$ and $\beta_{\max} = 20$. After training, the learned policy was tested for another $2,000$ episodes by setting $\beta = \beta_{\min}$, resulting in a greedy action-selection.[7]

5.1.3   Agent Performance Results

Summary statistics of the agents' performance during the $2,000$ test episodes after learning are listed in Table 3. Fig. 3 depicts the agents' performance relative to each region of the environment. Our goal was to design the agents such that, after training, their underlying characteristics and limitations would lead to contrasting behaviors. As we can see, the different characteristics indeed resulted in the agents attaining different performances in the task.

The optimized agent achieves the best performance among the three agents, reaching significantly higher levels. Compared to the high-vision agent, we see

---

[7]The performance of each agent *during learning* is detailed and discussed in B.

Table 3: The performance of each agent in $2,000$ test episodes. Legend: River: agent jumped to river; Car: agent was hit by a car; Time: maximum moves to pass a level (100) reached.

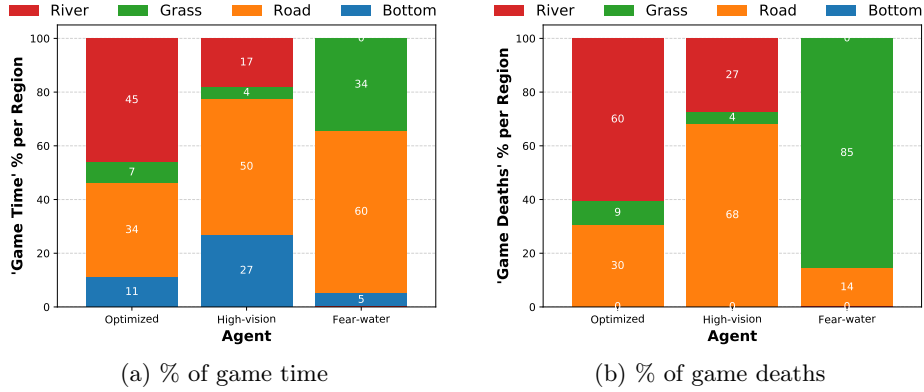| Agent	Mean Level	Number of Deaths				Mean Steps
		River	Car	Time	Total	
Optimized	4.6 ± 0.7	$2,395$	$1,051$	14	$3,460$	288 ± 26
High-vision	2.0 ± 0.4	411	888	$3,027$	$4,326$	291 ± 22
Fear-water	1.0 ± 0.0	0	$3,604$	$2,396$	$6,000$	222 ± 46

(a) % of game time

(b) % of game deaths

Figure 3: The percentage of time spent and deaths occurred at each environment region during $2,000$ test episodes. Legend: Grass: middle grassy strip; Bottom: grassy strip on the bottom.

that this is due to the differences in the vision range parameters. The high-vision agent can see cars at a higher distance, resulting in more cautious behavior while crossing the road. While this leads the agent to spend more time on the road, it gets hit by cars significantly less than other agents.

For the fear-water agent, receiving a highly negative reward from dying in the river (two times the magnitude of the lilypad reward) prevents it from reaching any lilypad. The agent was also not optimistically initialized, making it less motivated to explore and inadvertently discover the lilypads. As seen in Fig. 3, we see that the agent usually dies by getting hit by a car from the middle grass row. This is because the value of jumping into the river (action N) is significantly lower than that of any other action, *i.e.*, the agent "fears" falling into the water more than anything else.

5.2 Introspection and Explanation

Referring back to Fig. 1, an agent's history of interaction in our experiments comprises a total of $4,000$ episodes ($2,000$ training and $2,000$ testing). During

Table 4: Composition and budget of the summarization techniques used in the user study.

ID	Name	Summary Composition	k
1	*Maxima*	all observation maxima	4
2	*Minima*	all observation minima	4
3	*Certain*	all certain execution observations	4
4	*Uncertain*	all uncertain execution observations	4
5	*Frequent*	all frequent observations	4
6	*Infrequent*	all infrequent observations	4
7	*Transition-value*	half minima, half maxima	4
8	*Execution certainty*	half certain-execution, half uncertain	4
9	*Frequency*	half frequent, half infrequent	4
10	*All*	one of each type	6
11	*Sequence*	sequence from minima to maxima	1

that time, all the interaction data described in Sec. 3.1 was collected. We then applied our introspection framework to extract the interestingness elements in Table 12 for each agent.[8]

After introspective analysis, visual explanations were produced by generating summary video clips highlighting the agents' performances, according to the procedures described in Sec. 4. Because we were interested in capturing the agents' learned behavior and capabilities, the summaries were captured during the $2,000$ test episodes.[9] For the diversity metric $d(z_1, z_2)$ over highlights, we used the absolute difference of game points between the moments in which z_1 and z_2 were observed, the goal being to capture the agents' behavior at various stages of the game.

5.3 User Study

To assess whether the visual explanations derived from the different interestingness elements would help human users correctly understand the different agents' underlying capabilities and limitations, we conducted a user study. We recruited participants using the psiTurk tool [23] for Amazon Mechanical Turk (MTurk). A total of 82 participants (40% female, age group mode $[25 - 34]$ years) participated in the study, each receiving $7.50 for their completion of the Human Intelligence Task (HIT).

5.3.1 Summarization Techniques

Our objective was to understand the differential contribution of each interestingness element to understanding an agent's behavior. We wanted to know whether single elements are sufficient for correctly perceiving an agent's aptitude or if a combination works better to convey certain behavioral characteristics.

Table 4 lists the 11 *summarization techniques* used in our study. The first group of techniques $(1 - 6)$ generates summaries composed of highlights for a single type of interestingness element. The goal was to assess how each type exposes different characteristics of the agents' behavior. The second group of techniques $(7 - 9)$ generates videos composed of trajectories highlighting the extrema of each dimension of analysis, enabling us to evaluate the explanatory power of each dimension. The *All* technique (10), given a higher budget, generated highlights including 6 trajectories. The objective was to assess the effect of including one highlight per element. Finally, the *Sequence* technique (11) produces a single sequence highlight to determine whether a single trajectory is sufficient to reveal subtleties underlying the agents' behavior.

We used a trajectory length of $l = 21$ time-steps (≈ 5 secs.) for each highlight in a summary and a maximum length of 80 time-steps for sequence highlights. To avoid biases during the observation of highlights by human users, we removed all game information at the bottom of the screen from the videos, *i.e.*, the level, remaining moves, points and available lives.[10]

5.3.2 Conditions

Each condition of our study corresponds to a *scenario* where we assess the subjects' perception of all three agents' characteristics given one summarization technique. Since we have 11 techniques and three agents, we have two independent variables: *agent*, with 3 levels, and *scenario*, with 11. Informed by a pilot study, we determined that requiring subjects to watch videos for all players and all summarization techniques (a full within-subjects design), would be too onerous. On the other hand, getting accustomed to the task requires time and effort, which made exposing each subject to only one condition (between-subjects design) equally undesirable. Therefore, we opted for a partially-balanced, incomplete block experimental design [6], where each subject was exposed to 6 randomly-selected scenarios, and in each scenario subjects were exposed to the behavior of all three agents.

5.3.3 Experimental Procedure

After accepting the HIT, subjects started by agreeing to a consent form explaining the goals and protocol of the study. They were then redirected to a page
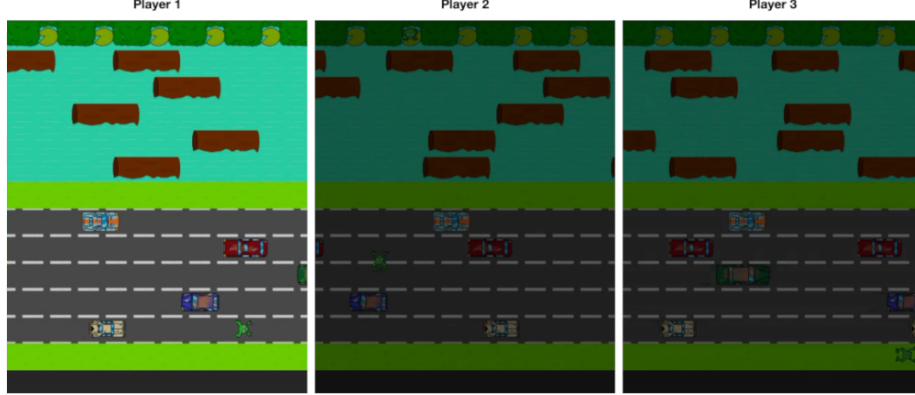
---

[8]B provides statistics on the interestingness elements captured for each agent.

[9]While introspection was performed over all the episodes, explanations were extracted only from the test episodes to avoid capturing mistakes due to the training process itself (exploration) rather than because of the agents' inherent limitations.

[10]The videos used in the study are at https://github.com/pedrodbs/frogger-study.

Figure 4: Questionnaire used in the user study.

describing the dynamics and rules of Frogger, after which they were presented with a 5-question quiz about the game. Failed questions were highlighted in red to reinforce the main concepts, but all participants were allowed to continue. The primary goal was to ensure that subjects understood the dynamics under-lying the task (*e.g.*, that reaching a new level speeds up cars and logs) rather than to prevent them for participating in the study.

An instruction page was then presented that explained what the participants had to do in each of the 6 scenarios. As seen in Fig. 4, each scenario showed videos of the three agents in parallel. The relative location of the video for each agent in the page (left, middle, right) was randomly assigned. Participants were told that they were watching videos of the gameplay of 3 different *players*, and that each scenario corresponded to a different set of players.

Subjects could start each video when they pleased and after playing all

videos (enforced programmatically), a questionnaire with the survey questions appeared below the videos, as depicted in Fig. 4. We designed a set of questions for each scenario to assess subjects' perception of the agents' underlying characteristics and also properties of their learned behavior, given the scenario's summarization technique.

As informed by the performance results presented earlier (see Fig. 3), the agents' capabilities in each region are quite different. By observing their learned behavior, we see that each region in Frogger provides different challenges and requires distinct learned skills to successfully overcome them. As such, we created two *region-related* questions:

Time: *"Select the regions where you think each player normally spends more time playing the game (select at least one)".*

Practice: *"Select the regions where you think each player needs more practice (select none if you think the player doesn't need more practice)".*

These questions assess the subjects' perception of the differentiated behavior of agents in each of the environment's regions. *Time* assesses relative presence while *practice* evaluates difficulties experienced in each region. For each question subjects could select from {*river, middle grass row, road, bottom grass row*}. The outcome of each question is thus a Boolean variable for each region.

We also designed two questions about the overall *aptitude* of the agents:

Level: *"The player is capable of reaching advanced levels".*

Help: *"The player needs help to be better at the game".*

Subjects were asked to rate their agreement to each statement on a 5-point Likert scale (1 strongly disagree – 5 strongly agree). The goal was to evaluate the subjects' perception of the overall capabilities of each agent. In particular, *level* estimates the perception of how well an agent plays the game, while *help* evaluates whether subjects could correctly determine if an agent needed intervention to improve its performance. Subjects were instructed to answer questions with respect to what they inferred about the players given what they saw in the videos clips, rather than evaluating the player solely on their performance in the videos.[11] A final question assessed the subjects' *confidence* in their responses on a 5-point Likert scale (1 not confident – 5 very confident). The outcomes of these questions are thus ordinal variables with values in $[1, 5]$.

After answering the questions for all 6 scenarios, a final page collected subjects' demographics and general opinions (open question) about the study. Subjects then returned to AMT to collect their compensation.

6 Analysis and Results

We are mainly interested in answering these two research questions:

---

[11] Subjects were allowed to play, stop and rewind videos as many times as they wanted.

Table 5: Significant differences in the responses of the region-related questions. Legend: V: Cramér's V statistic; Scen.: scenarios with the most pairwise significant differences. Empty cells denote not significant differences ($p \geq 0.01$).

Response variable	Agent	River		Grass		Road		Bottom	
		V	Scen.	V	Scen.	V	Scen.	V	Scen.
Time	Optimized	0.52	2			0.50	2	0.19	
	High-vision	0.65	2, 4, 5			0.42	4	0.69	2
	Fear-water			0.41	2, 4	0.36	1	0.42	4
Practice	Optimized	0.30	6	0.26		0.46	11, 3		
	High-vision			0.28	6	0.49	6	0.27	6
	Fear-water					0.35	2		

RQ1: *Does the information conveyed by the several summarization techniques induce a different perception of each agent's aptitude?*

RQ2: *What summarization techniques enable a correct perception of the agents' aptitude in the task?*

To address these questions, we started by removing the responses of subjects who did not seem to take the task seriously. To that end, we used the number of *correct answers* in the Frogger quiz and the *time* taken to answer the survey. Subjects answered on average 4.2 ± 1.0 quiz questions correctly (out of 5), and took an average of 19.5 ± 11.9 min. to complete the survey. We removed from further analysis the responses of subjects who either had 2 or fewer correct answers or spent less than 6 min. on the survey ($<$ 1 min. per scenario), thereby eliminating 5 subjects. The data for the remaining 77 subjects resulted in each scenario being sampled 42 ± 4 times (min: 36, max: 50).

6.1 Analysis of Region-Related Questions

6.1.1 Research Question 1

To address RQ1 for the region-related responses (Boolean variable for each region), we first grouped responses for each region by agent and then by scenario (summarization technique). A χ^2 (chi-squared) test was then performed over the sum of positive responses. The effect size was calculated using Cramér's V statistic with the bias correction in [5]. This lets us assess whether different techniques induced different perceptions of the agents' behavior in the several regions. A Bonferroni correction post-hoc comparison was then performed to examine significant pairwise differences.

Table 5 shows the significant effects ($p < 0.01$) of the summarization technique on the subjects' responses found for each agent.[12] We also show which techniques had the most pairwise significant differences, as informed by the

---

[12]C contains plots depicting subjects' responses for the region-related questions.

Table 6: Pairing of response variables and agent performance data (ground-truth). Means refer to performance measured across the 2,000 test episodes.

Response variable	Agent performance variable
Time	mean percentage of game time spent in the region
Practice	mean number of deaths occurred in the region (excludes lives lost due to maximum number of moves achieved)

Bonferroni comparison—this indicates what interestingness elements and introspection dimensions exposed the most distinct characteristics of the agents.

For the *time* response, significant effects were found for all agents in the road and bottom grass regions. Because the agents spend most of their time in these regions (see Fig. 3a), the behavior captured by the different techniques is quite diverse, leading to different interpretations of each agent's characteristics. For example, seeing a frog successfully avoid or get hit by a car can lead to different expectations of the time spent in that region. For the river and middle grass regions, subjects' responses lead to different results. The optimized and high-vision agents can quickly cross the middle grass region as soon as a log is available, contrasting to the fear-water agent who spends most of its time in this region. Thus, as expected, significant differences were found for the optimized and high-vision agents in the river, while the fear-water agent induced different perceptions of behavior only in the middle grass region. Overall, the most significant pairwise differences were produced in scenarios 2 (observation minima) and 4 (uncertain execution), denoting the importance of exposing "hard" situations to be able to assess an agent's relative presence in different regions.

The results for the *practice* response variable were different but consistent with how the agents lose lives in different regions, as depicted in Fig. 3b and Table 3. Significant effects were found for all agents only in the road region. This is where most of deaths occur, making the other regions less sensitive to the difficulties experienced by the agents, as captured by the various techniques. Overall, the most significant pairwise differences occurred in scenario 6, which exposes an agent's behavior in infrequent situations. This was particularly important for the interpretation of agent high-vision's behavior in all regions except the river, since those are where it spends most of its time.

6.1.2 Research Question 2

To address RQ2 for the region-related questions, we first paired each response variable with a game performance variable, as listed in Table 6. The performance data serves as ground truth for the agents' behavior and characteristics against which we compare the subjects' interpretations to identify which summarization techniques allowed for a correct assessment of the agents' aptitude in each region.

Then, for each response variable, we transformed the vector containing the

Table 7: JSD between response and performance variables for the region-related questions, for each scenario (ID). Legend: **blue:** low divergence; **red:** high divergence.

ID	Time			Practice		
	Optim.	High-vis.	Fear-wat.	Optim.	High-vis.	Fear-wat.
1	0.09	**0.70**	0.60	**0.01**	0.47	0.54
2	0.45	0.60	**0.03**	0.35	0.08	**0.80**
3	0.18	**0.70**	0.15	**0.02**	0.63	0.51
4	0.37	**0.04**	0.18	0.29	0.18	0.61
5	**0.03**	0.55	0.07	**0.02**	0.16	0.60
6	**0.05**	**0.77**	0.57	**0.00**	**0.75**	**0.73**
7	0.11	0.33	0.35	0.47	**0.03**	**0.66**
8	0.18	0.14	0.13	0.35	0.23	**0.74**
9	**0.03**	0.30	0.50	**0.01**	**0.04**	0.44
10	0.22	0.11	0.18	0.36	0.15	0.61
11	**0.01**	0.37	0.27	**0.01**	0.32	0.62

percentage of subjects' responses across all regions, for each agent and summarization technique, into a Boltzmann distribution.[13] Similarly, we derived a Boltzmann distribution from the mean values of the corresponding performance variable, resulting in the expected relative performance of a given agent in each region.[14] The Jensen-Shannon divergence (JSD) [10] between the performance (ground-truth) and each response distribution was then computed to determine whether the subjects' responses diverged from the agents' actual performance and to identify the agents for which the distributions diverged the most.

Table 7 presents the results of this analysis. We highlight the summarization techniques that resulted in either a very correct (low divergence) or very incorrect (high divergence) interpretation of the agents' performance relative to each region. For the optimized agent, a correct assessment for the *time* response variable was achieved in scenarios 5, 6, 9 and 11. A similar result was seen for the *practice* response variable, denoting the importance of the frequency dimension for correctly assessing where an agent spends most of its time and where it needs more (or less) practice.

Regarding agent high-vision, due to its perceptual limitations related to seeing cars, the uncertain situations (scenario 4) occur mostly on the road, leading to the most correct interpretation of where its time is spent. Importantly, exposing subjects only to maxima, certain and infrequent situations (scenarios 1, 3 and 6, respectively) led to a very incorrect assessment, leading subjects

---

[13]*E.g.*, if all subjects selected all regions in the *practice* question for an agent in a given scenario, this means they believed that the agent needed to practice equally in each region, resulting in a probability of 1/4 for each element of the response distribution.

[14]*E.g.*, for the *practice* response variable, this corresponds to the probability of an agent losing a life in each region.

Table 8: Significant differences ($p < 0.01$) in the responses of the aptitude-related questions. Legend: ϵ^2: epsilon-squared effect; Scenario: scenarios with most pairwise signif. differences.

Response var.	Agent	ϵ^2	Scenario
Level	Optimized	0.32	2
	High-vision	0.39	2, 4
	Fear-water	0.07	4
Help	Optimized	0.33	2, 4
	High-vision	0.40	2, 5
	Fear-water	0.07	4

to believe that the agent either spends much more or much less time in each region than it really did. For the *practice* question, the transition-value and frequency dimensions (scenarios 7 and 9) led to the most correct perception. These techniques provide a balance of "good" and "bad" moments of performance.

The fear-water agent produced rather different results. The agent spends most of its *time* in the middle grass row, thus only the observation minima (scenario 2) which all occurred in that region led to a correct perception. For *practice*, the frequency dimension (scenario 6) achieved the best results but no technique resulted in a very correct interpretation of where the agent loses its lives. This is due to the agent's motivational limitations, which leads all summarization techniques to capture only situations where it performs poorly, giving the impression that the agent needs practice in all the regions equally when in fact it has the most difficulty reaching the river (middle grass region).

6.2 Analysis of Aptitude-Related Questions

6.2.1 Research Question 1

To address RQ1 (differences in the subjects' responses) for the aptitude-related questions (ordinal variables in $[1 - 5]$), we first grouped responses by agent and then by scenario. We modeled the response for each agent as an ordinary least squares (OLS) regression model and performed a Kruskal-Wallis H-test. This informs about whether different summarization techniques resulted in different perceptions of the agents' aptitudes. Effect sizes were calculated using the ϵ^2 (epsilon-squared) statistic described in [32]. This was followed by a Bonferroni correction post-hoc pairwise comparison.

Table 8 shows a significant effect of the summarization technique (scenario) on the *level* and *help* responses for all agent types ($p < 0.01$). The results for effect size also show that the videos depicting the behavior of optimized and high-vision agents had a greater impact on the differences induced by the summarization techniques. This is expected since these agents exhibit more diverse behavior compared to the fear-water agent.

Table 9: Pairing of response variables and agent performance data (ground-truth).

Response variable	Agent performance variable
Level	mean game level achieved
Help	mean number of lives lost per game

The Bonferroni comparison shows that scenarios 2 (infrequent) and 4 (uncertain execution) had the greatest impact on inducing different responses from the subjects. This indicates that the behavior exhibited by the agents during "hard" situations are qualitatively distinct and lead to very different assessment about their overall aptitude in the task.

6.2.2 Research Question 2

RQ2 investigates the correctness of the subjects' assessment of the agents' capabilities and limitations in different scenarios. We again paired each response variable with a corresponding agent performance variable, as listed in Table 9. First, we linearly rescaled (min-max scale) the response variables for all scenarios and the corresponding performance variables so that they fell in the $[0, 1]$ interval, and then modeled them as Gaussian distributions. For example, for the *level* performance variable, this models the probability of an agent reaching high levels. Then, for each agent, the Hellinger distance (H) [17] between the distributions of the response and performance variables was calculated for each scenario. This resulted in a divergence measure in $[0, 1]$ whose interpretation is that techniques for which H is close to 1 did not allow for a correct understanding of the underlying aptitude, while a low H means a correct interpretation.

Table 10 shows the results of our analysis by listing the Hellinger distance between the response and game performance variables for each agent and summarization technique. The most noticeable result concerns the fear-water agent, where the subjects' perception of the agent's capabilities (ability to reach high levels) and its limitations (average number of lives lost) was incorrect, regardless of the summarization technique used. On one hand, the behavior exhibited by the agent due to its "motivational impairments" is so limited that no interestingness element captures qualitatively distinct characteristics. On the other, its aptitude was greatly overestimated when compared to the other agents.[15]

In contrast, the results for the other agents show that some techniques might be better correctly identifying an agent's overall capabilities. Specifically, for the high-vision agent, scenarios 1, 3, 6 and 11 resulted in the highest H distances. Techniques 1 (maxima), 3 (certain execution) and 11 (likely sequences to a sub-goal) contain elements denoting moments where the agent performs

---

[15]Because we asked subjects to imagine the players' *usual* performance when responding to the questionnaire, they might have overestimated agent fear-water's capabilities, *e.g.*, by imagining that it could eventually cross the river and reach the lilypads.

Table 10: *H* distance between response and performance variables for the aptitude-related questions, for each scenario (ID). Legend: **blue:** low divergence; **red:** high divergence.

ID	Level			Help		
	Optim.	High-vis.	Fear-wat.	Optim.	High-vis.	Fear-wat.
1	0.27	**0.95**	**0.99**	0.31	0.57	**0.99**
2	0.56	0.48	**0.99**	0.40	0.30	**0.99**
3	0.45	**0.91**	**0.99**	0.41	0.50	**0.99**
4	0.47	0.52	**0.99**	0.37	0.20	**0.98**
5	0.32	0.53	**0.99**	0.26	0.28	**0.99**
6	0.39	**0.96**	**0.99**	0.11	0.63	**0.99**
7	0.31	0.73	**0.99**	0.14	0.27	**0.99**
8	0.35	0.71	**0.99**	**0.04**	0.20	**0.99**
9	0.31	0.71	**0.99**	**0.04**	0.26	**0.99**
10	0.38	0.75	**0.99**	**0.09**	0.32	**0.99**
11	0.40	**0.91**	**0.99**	0.36	0.55	**0.99**

well in the task, which could have led subjects to overestimate the agent's capabilities compared to the optimized agent. On the other hand, in scenario 6 (infrequent) subjects were shown situations where the agent fails, leading to an underestimation of its relative aptitude. In contrast, scenarios 2 (minima), 4 (uncertain) and 5 (frequent) allowed for a more accurate assessment, particularly of the agent's limitations.

For the optimized agent, contrasting to the fear-water agent, the results show that subjects might have underestimated its performance when observing highlights of "hard" situations, such as in scenarios 2 (minima) and 4 (uncertain execution). Scenario 3 (certain execution) attained also a high JSD score for the agent, probably because it included situations highlighting the agent's skills in jumping on logs, which is how the agent loses most of its lives. Regarding the techniques supporting a more correct understanding of the agent's capabilities, for the *level* response this was achieved in scenario 1 (maxima), while for *help* it occurred in scenarios 8 (frequency) and 9 (execution certainty). Overall, these techniques highlight the agent's ability to achieve its goals and overcome difficult situations, thus correctly showing that, in general, it does not need much help and does not lose many lives.

6.2.3   Assessing Relative Performance

The results in the previous section indicate, for some scenarios, a lack of agreement between subjects' responses and the agents' game performance. However, this might be due to our experimental design, where each scenario presented highlights for the three agents simultaneously. As such, it is also worth analyzing which summarization techniques resulted in a correct perception of the
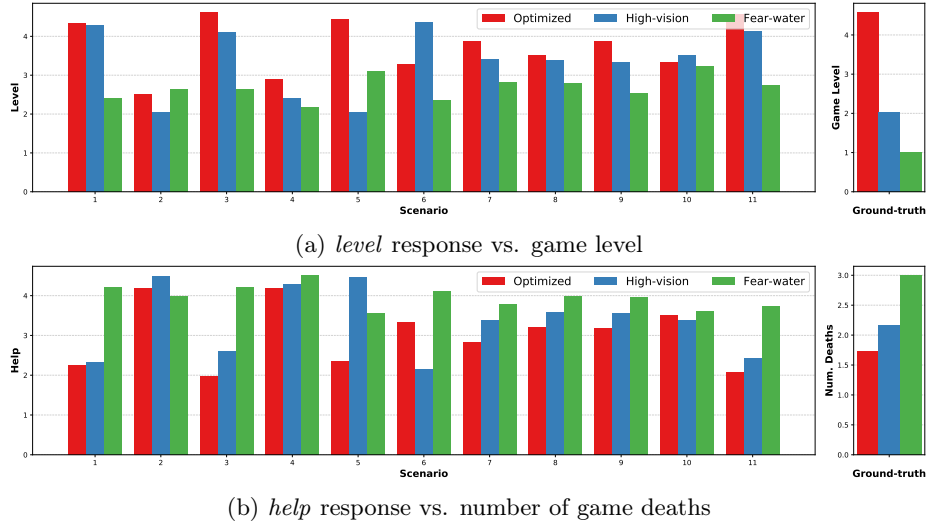
(a) *level* response vs. game level

(b) *help* response vs. number of game deaths

Figure 5: Comparison of subjects' mean responses for each agent and summarization technique (Scenario $1 - 11$, left) and the agents' game performance across the $2,000$ test episodes (right). We note the different scales between responses and ground-truth data.

relative performance between the agents.

Figure 5 helps illustrate the idea behind our approach. By comparing the shape of the individual scenario plots to the ground truth plot, we see that some techniques appear to support a correct perception of the agents relative abilities, while others do not. To quantify this difference, we compared the distribution of subjects' responses across the three agents with the distribution of their actual performance. Specifically, for each scenario, we derived Boltzmann probability distributions for the agents from the mean values of the responses and the corresponding performance variables. We then computed the JSD between each response distribution and the ground-truth performance distribution.

The results are listed in Table 11, where the *diverging agents* column shows which agents are responsible for the distributions being divergent when JSD ≥ 0.5. For the *level* response variable, we see that only showing highlights where the agents perform their best (scenario 1) does not lead to a completely correct perception of their capabilities. Meanwhile, showing only observation minima (scenario 2) and infrequent (scenario 6) situations provides an incorrect impression of the agents' relative abilities—the 'failing' behavior of the fear-water agent was perceived to be very similar to that of the other two agents. Scenario 10, by including moments generated by means of all interestingness elements, appears to give subjects the idea that all agents have a similar level of performance (see Fig. 5a). Overall, frequent situations (scenario 5) led to the most accurate perception of the agents' relative performance.

For the *help* response variable, we observe a similar result for scenario 2:

24

Table 11: JSD between relative response and performance for the aptitude-related questions, for each scenario (ID). Legend: **blue:** low divergence; **red:** high divergence.

ID	Level		Help	
	JSD	Diverging agents	JSD	Diverging agents
1	0.28		0.03	
2	0.51	Fear-water	0.53	High-vision, Fear-water
3	0.09		0.02	
4	0.04		0.21	
5	0.00		0.73	High-vision, Fear-water
6	0.92	Optimized, High-vision	0.03	
7	0.10		0.04	
8	0.23		0.06	
9	0.06		0.06	
10	0.52	High-vision	0.24	
11	0.12		0.02	

showing the agents' behavior in hard situations leads to the incorrect perception of all agents needing help equally. For the frequency elements, we see an effect opposite to the *level* response: showing only frequent situations (scenario 5) did not result in correctly perceiving the help needed by agents when compared to the mean number of deaths, while showing only infrequent situations (scenario 6) did. Overall, the frequency dimension (scenario 9) seemed to strike the right balance, resulting in a low JSD score for both response variables. Other techniques that resulted in a correct perception of relative help were 1 (maxima), 3 (certain execution), 7 (both minima and maxima) and 11 (likely sequence to sub-goal). As such, unlike the *level* response, subjects were able to establish a correct relative assessment of the agents' limitations by seeing moments in which they performed their best.

6.3 Confidence Analysis

Finally, we wanted to determine whether summarization techniques influenced the subjects' confidence response. A Kruskal-Wallis analysis of variance did not reveal a significant effect of the summarization technique (scenario) on the subjects' *confidence* response ($p = 0.047$). Fig. 6 shows the mean responses per scenario. We see that, in general, the subjects' confidence in their answers was not greatly affected by exposure to particular scenarios. A Bonferroni correction post-hoc comparison revealed that the largest difference was found between scenarios 11 (most confident) and 10 (least confident) ($p = 7.5 \times 10^{-4}$, $\varepsilon^2 = 0.34$). This further supports both the usefulness of sequence highlights (11) and that including moments from all interestingness elements can confound subjects and lead to an incorrect assessment of their aptitude.
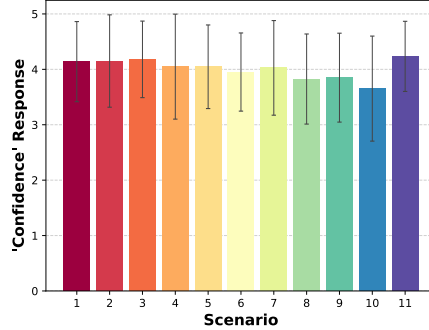
Figure 6: The subjects' mean confidence in their responses in each scenario.

6.4 Discussion

In this section we discuss in depth the results of our experimental study and provide the main insights stemming from using each summarization technique.

6.4.1 Usefulness of the Summarization Techniques

Based on our results, a main conclusion is that no single technique can induce a correct understanding of the underlying capabilities and limitations of *all* agents in *all* situations. Some techniques seem better at helping a user understand an agent's limitations and identify opportunities to intervene, others at identifying the subtasks on which an agent fails or performs the best. Further, some elements enable a correct perception of the relative performance between different agents, while others lead to incorrect comparisons. As our results show, different agents required exposing different aspects of their interaction for users to have a correct understanding of their aptitude.

A related insight is that the summarization techniques, and the associated interestingness elements, are only as good at helping capture different characteristics of RL agents as the richness of the agents' underlying capabilities. Specifically, agents that have a predictable, monotonous performance—*e.g.*, the fear-water agent that had poor performance throughout—may not generate the necessary numerical nuances targeted by our introspection framework. For example, if the values associated with the observations are all high, one cannot distinguish between observation minima and maxima. Likewise, if the observed behavior is highly deterministic, then uncertain elements cannot be determined.

Regarding the results of the region-related questions, note that the relative performance of an agent in the different regions of Frogger can be associated with solving different challenges, or *subtasks*. Our results show that assessing an agent's aptitude in each subtask can be hard and ultimately depends on the agent's overall performance. For high-performing agents, the maxima, cer-

26

tain execution, frequent, infrequent,[16] and sequence elements capture moments where they excel, resulting in a correct perception of how quickly the agent addresses each subtask and whether it needs adjustments. In contrast, for agents that perform poorly overall, situations captured by the minima and frequent elements can be good indicators of the relative amount of time spent addressing each subtask, but can lead users to incorrectly believing that the agent requires improvement in all subtasks equally.

For agents with perceptual limitations, the uncertain execution element reveals situations where they are unsure of what to do, which can lead to a correct identification of the subtasks that may take more time to be addressed. In addition, the transition-value and frequency dimensions, by providing a balance between situations, can provide the best understanding of the subtasks for which such agents need further adjustment.

Regarding the analysis of the subjects' perception of the relative aptitude between agents, our results show that exposing only "good" ("bad") moments can lead users to incorrectly perceiving aptitude. In particular, exposing users to moments where agents perform their best (observation maxima and certain execution) can lead to an overestimation of their abilities, while showing only moments where the agents succeed or fail in hard situations (observation minima and uncertain execution) can lead users to incorrectly perceive relative agent limitations and to believe that all agents fail equally. The problem is that by providing examples only of where an agent excels or where it fails, users may not understand the parts of the task that required more of the agent's attention (time spent), nor where and how the agent needs to be improved.

One dimension of analysis providing good overall results was frequency. On one hand, the results show the importance of frequent elements for correctly determining where an agent spends most of its time. On the other, infrequent situations are also relevant in that they can help identify when the agent fails—a proficient agent might solve an infrequent challenge quickly, while a limited agent might spend too much time solving it or require human intervention. The region-related responses show that the frequency dimension, by providing a balance between common and rare situations, provides a good indication of the improvements needed in each subtask. In conclusion, frequency seems to be useful for showing an agent's overall behavior characteristics, while the transition-value dimension and other more complex elements relying on the learned value function help reveal its difficulties and capabilities in hard, challenging situations.

Another technique worth mentioning is likely sequences to sub-goals, which involves capturing trajectories of an agent going from a difficult situation (local minimum) to a learned goal (local maximum). An advantage of this technique is that an observer gets to see the "normal" behavior of the agent in the task, which is related to its learned strategy. Our results show that even with a single trajectory, subjects could correctly perceive the unique characteristics of the agents' behavior and infer their underlying relative aptitude. Although this technique did not reach the lowest divergence scores overall, sequences may

---

[16]Infrequent situations help highlight even more the robustness of a good policy.

enable a reasonably accurate assessment of how the agent performs in the task, particularly when given a limited explanation budget (time).

Our results also captured an interesting result regarding the *all* technique that creates summaries containing one highlight for each interestingness element. One might think this technique would be optimal, given that it highlights all the different aspects of the agent's interaction. However, in our study it led to the lowest confidence in responses and poor perceptions overall. We believe that subjects may have been confused by the diversity of moments highlighted by each element, making it difficult to determine an agent's overall aptitude. Also, half the highlights show infrequent situations or where an agent is uncertain of what to do, leading subjects to underestimating its performance.

6.4.2 Mitigating Users' Incorrect Expectations

A problem with visual explanations based on highlights is that, by definition, they do not capture all aspects of an agent's interaction with the environment. Users need to extrapolate the behavior of an agent beyond what they see, a problem that can be exacerbated by expectations users have regarding an agent's performance in the task. In our study, subjects' had prior beliefs about how "normal" players behave in Frogger, likely leading to incorrect perceptions of aptitude in some scenarios. Also, by design, our experiment involved visualizing agents simultaneously, which could have led to the creation of false expectations based on the best and worst performance observed among all agents.

Our results show that some summarization techniques do lead to a correct comparative assessment between different agents. However, we conclude that, in general, highlights should be accompanied by more quantitative measures about the agents' overall performance in the task. This may be necessary for users to form a fair judgement about the nature of the agents' capabilities and limitations; and of when and how they should intervene. For example, observing a high-performing agent failing to perform a particular subtask can direct a user to correct specific aspects of the agent's design. Similarly, such information could help users understand that not all agent limitations are the same and that different agents may require distinct adjustments.

Another solution to mitigate problems of false expectations is to provide users with a description of the information the highlights are conveying about an agent's behavior. This would be particularly useful in our framework given the variety of interestingness elements and the qualitatively distinct aspects of an agent's interaction that each captures. Therefore, users could better understand agents' limitations and set more realistic expectations about their behavior.

6.4.3 Overall Discussion

A few general notes about the proposed framework are worth mentioning. First, our framework is *domain-independent*, meaning that the data that is collected and analyzed is agnostic to the specific learning scenario. In addition, it is

algorithm-independent in that it can be used in conjunction with standard RL tabular methods without having to modify the learning mechanism itself.

We note also that the dimensions of analysis proposed here provide mathematical interpretations of different *concepts* for analyzing an RL agent's interaction with the environment. Namely, each concept provides different insights over the agent's behavior that are useful to judge its performance and to understand how it can be improved. Importantly, we do not claim that the list presented in this paper is complete or definitive—other aspects of an RL agent's interaction can be defined, and different interpretations, possibly applicable to different RL algorithms, can be proposed for the same concepts.

We chose to develop our initial framework using low-dimensional state spaces because, given the inherent complexity of sequential decision-making, it is important to first determine how to generate the critical *elements* for explainable RL. For that purpose, the concepts behind our elements—and their effects in enabling users to understand the aptitude of agents—are more important than the particular evaluation domain or the RL architecture used to train the agents. Furthermore, we note that the subjects of our study were unaware that the players were in fact autonomous agents, let alone that RL was used to derive their behavior. As such, our evaluation of the usefulness of the explanation concepts is agnostic to the underlying representations and RL algorithm used.

7   Related Work

We now review recent works within XRL that have addressed some of the problems identified in this paper. We summarize their technical approaches, highlight their main contributions and compare them to our framework.

7.1   Deriving Language Templates

One approach is to use language templates to translate elements of the problem or the agent's policy into human-understandable explanations. For example, Khan et al. [20] used the discounted occupancy frequency according to a policy to form contrastive explanations about the expected return in a given state by executing the optimal action versus alternative ones. In particular, their minimal sufficient explanations (MSE) populate templates and convey to the user the probability or frequency of reaching certain states with high rewards by using the optimal action but not other actions. However, explanations can be long and not suited for non-technical users, for whom the concept of utility may not be easy to understand.

Elizalde et al. [9] developed an intelligent assistant for operator training (IAOT) relying on factored state scenarios. The system is capable of providing explanations using templates tailored to novice, intermediate, and advanced users. Explanations are generated whenever a mistake is made and include the optimal action that should be performed, a justification in natural language according to the user level, and a (domain-dependent) visual diagram highlight

the most relevant variable. The latter corresponds to the state feature providing the highest positive difference in the value function. A drawback of this method is that it requires creating a knowledge base for the justifications in the templates, *e.g.*, obtained from written documentation or domain experts.

Wang et al. [34] generate explanations for POMDPs that expose users to the agent's policy, beliefs over states of the environment, transition probabilities, rewards, etc. in human-robot interaction (HRI) scenarios. The agent can refer to properties of the environment and its behavior but cannot identify the most pertinent situations or justify its actions.

Unlike these approaches which require a large amount of domain knowledge to create explanations, our framework relies on standard RL data and is domain-agnostic. In addition, techniques like ours that use visual explanations of behavior make explanation more accessible to non-expert users.

7.2   Deep RL Over Image-Based Inputs

Other approaches aim at providing insights about policies learned using deep RL techniques on image-based input tasks. For example, Zahavy et al. [36] aggregate the high-dimensional state space of ATARI games into a hierarchical structure using t-Distributed Stochastic Neighbor Embedding (t-SNE) for dimensionality reduction. A graphical user interface then allows the manual selection and visualization of the aggregation of different states according to several types of features. However, the representation and provided information make it difficult for non-RL experts to extract meaningful information.

Greydanus et al. [12] use perturbation-based techniques to identify image regions that result in significant policy changes. The result is a video of agent performance where the relevant regions of the image are overlaid with different colors. This approach is computationally demanding as it requires sweeping whole input images at every step, and users of the system would be required to observe full-length videos of an agent's performance to understand its policy.

An advantage of these approaches is that they allow scaling to large input space domains. A drawback is that they cannot generalize to tasks with non-image-based inputs. Our approach is agnostic to the type of observations to derive the different elements. In addition, unlike these approaches that expose the entirety of an agent's behavior, we follow others who select only the moments of the agent's behavior that are relevant according to different criteria.

7.3   Abstracting Explainable Representations

Koul et al. [21] learn finite state representations of policies learned using deep recurrent neural networks (RNNs) that are agnostic to the input type. The idea is to abstract the learned policy into a simpler, more interpretable structure. First, quantized bottleneck networks (QBNs) are learned from a trained RNN policy that can reconstruct the observations and hidden states using a lower-dimensional representation space. From this, a finite state machine is extracted and discretized. Although the resulting structures provide insights on the nature

of the learned policy, *e.g.*, that it contains certain cycles, it is hard to interpret the semantics behind the discretized observations and hidden states.

Other works focus on abstracting state representations of the task and creating graph structures denoting the agent's behavior. For example, Hayes and Shah [16] and van der Waa et al. [33] frame explanations as summaries assembled from outputs of binary classifiers that characterize the agent's trajectories and decisions. The system in [16] was designed for HRI scenarios where users pose questions that are identified from a set of question templates, resolved to a relevant set of world states, and then summarized and composed into a natural language form to respond to the inquiring user. The approach in [33] allows users of asking contrastive questions (foil) that may contradict the policy learned by an RL agent. Similar to our element of sequences, a path can be constructed that includes the most likely state-action pairs. An explanation is then constructed from a language template.

In contrast to these approaches, our framework identifies relevant aspects of the agent's behavior according to different criteria. Each element provides different insights about the agent's learned policy and provides the semantics for the visual explanations.

7.4 Identifying Key Interaction Moments

Approaches more similar to ours try to identify key moments of the agent's interaction. For example, Huang et al. [18] proposed a system capturing *critical states*, in which the value of one action is comparatively much higher than others. A user study was performed where subjects were exposed to critical states extracted for two different policies in the game of Pong, where one had a much higher training time. Subjects were then asked whether they trusted each policy and whether they would take control over them in a series of query states. Results show that exposing users to critical states enables an appropriate level of trust with regard to the agents' performance.

Our approach for visual explanations is based on the highlights system proposed by Amir and Amir [3]. Their system captured video clips highlighting an agent's behavior in the game of Pacman using the concept of importance, dictated by the largest difference in the Q-values of a given state. This allows the identification of trajectories that are representative of the agent's best performance. A user study was conducted where subjects were exposed to highlights of three agents with varying training times—and hence different performance levels in the task. Their results show that the proposed importance metric allowed a correct identification of the agents' relative performance when compared to other baseline summarization methods.

A first observation is that we go beyond the concept of importance for explaining the agent's behavior proposed in these papers as dictated solely by the highest value-difference states. In particular, we propose mathematical interpretations within RL for concepts such as frequency, uncertainty, predictability and contradiction. Our framework also includes an element that allows visualizing relevant sequences of states rather than single moments. With respect

to the user studies, the previous works mainly focused on assessing the perception of how good agents are (or whether users should trust them based on their performance). In this paper we were mainly concerned with establishing connections between the different elements and the correct understanding of both the agent's capabilities and limitations. Importantly, our results provided insights on which elements are better at indicating when and how agents might need intervention to improve their performance.

8 Conclusions and Future Work

In this paper we introduced a framework for introspective XRL. The approach first performs introspection by analyzing an agent's history of interaction with its environment, extracting different interestingness elements each capturing a particular aspect of the agent's experience. We showed how to generate visual explanations from these elements in the form of video-clips highlighting diverse moments of the agent's experience while using its learned policy.

We then showed the results of an experimental study using the video game of Frogger. First, we created agents with different perceptual capabilities and motivations to simulate potential problems in agent designs. We then applied RL to each agent, resulting in different behaviors in the task and degrees of performance, as measured through various criteria. We conducted a user study where we presented subjects with videos showing the agents' performance that were created by our explanation framework according to different summarization techniques. Each technique used different interestingness elements or a combination of them, to generate the video highlights for each agent.

Overall, our results show that no single summarization technique—and hence no single interestingness element—provides a complete understanding of every agent in all possible situations of a task. Ultimately, a combination of elements enables the most accurate understanding of an agent's aptitude in a task. Different combinations may be required for agents with distinct capabilities and levels of performance. The results also show that the framework is useful for assessing the behavior of agents with different levels of proficiency in various parts of a task—the nuances of their behavior can be successfully captured by the different interestingness elements.

Our results highlight in particular the element of frequency, which provides a good understanding of an agent's overall behavior characteristics. As for the transition-value dimension, it can denote difficulties and capabilities of agents in hard, challenging situations. In addition, the element producing the most likely sequences to subgoals can provide a short but accurate understanding of an agent's learned strategy. Another insight stemming from the results is that combining many types of elements in a single video summary can confound users. We suggest that supplementing video highlights with information about the elements used to produce them and agents' overall performance could mitigate issues with users setting incorrect expectations about agent aptitude.

Our introspection framework was designed to be used through the different

stages of an agent's lifecycle and for different explanation modes. For example, it can be used: *during learning*, to track the agent's learning progress and acquired preferences or *after learning*, to summarize the most relevant aspects of the interaction; *passively*, to let a user ask the agent about its current goals or to justify its behavior in any given situation or *proactively*, to let the agent consult the user in situations where its decision-making is more uncertain or unpredictable. We are currently developing a platform where the competency of RL agents, including competency in learning, can be assessed for different tasks, and where users can query about an agent's abilities in novel situations.

We are also currently extending the introspection framework to support analyzing high-dimensional domains that require the use of deep-RL techniques by defining interpretations of the concepts behind the different interestingness elements that can leverage deep network representations. Some elements allow for a direct translation to deep-RL architectures, *e.g.*, estimating execution certainty from the output distribution of a policy network or transition value from situations where executing different actions leads to strictly increasing or decreasing value. Others may require more specialized structures, *e.g.*, the transition certainty of a situation can be estimated through generative models [*e.g.*, 13, 14] that predict the likelihood of next states, while a model for characterizing the familiarity of inputs [*e.g.*, 2] can be used to determine the frequency of observations.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. ISBN 1-55860-153-8.

[2] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In C. V. Jawahar, H. Li, G. Mori, and K. Schindler, editors, *Computer Vision – ACCV 2018*, pages 622–637, Cham, 2019. Springer International Publishing. ISBN 978-3-030-20893-6.

[3] D. Amir and O. Amir. Highlights: Summarizing agent behavior to people. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 1168–1176, Richland, SC, 2018. IFAAMAS.

[4] R. J. Bayardo, Jr. and R. Agrawal. Mining the most interesting rules. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 145–154, New York, NY, USA, 1999. ACM. ISBN 1-58113-143-7. doi: 10.1145/312129.312219.

[5] W. Bergsma. A bias-correction for Cramér's V and Tschuprow's T. *Journal of the Korean Statistical Society*, 42(3):323 – 328, 2013. ISSN 1226-3192. doi: 10.1016/j.jkss.2012.10.002.

[6] R. C. Bose, W. H. Clatworthy, and S. S. Shrikhande. Tables of partially balanced designs with two associate classes. Technical report, North Carolina State University. Dept. of Statistics, 1953.

[7] M. M. A. De Graaf and B. F. Malle. How People Explain Action (and Autonomous Intelligent Systems Should Too). Technical Report FS-17-01, AAAI 2017 Fall Symposium on Artificial Intelligence for Human-Robot Interaction, 2017.

[8] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, Dec 1959. ISSN 0945-3245. doi: 10.1007/BF01386390.

[9] F. Elizalde, E. Sucar, J. Noguez, and A. Reyes. Generating explanations based on markov decision processes. In *MICAI 2009: Advances in Artificial Intelligence*, pages 51–62, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-05258-3. doi: 10.1007/978-3-642-05258-3_5.

[10] B. Fuglede and F. Topsoe. Jensen-Shannon divergence and Hilbert space embedding. In *Proceedings of the International Symposium on Information Theory, 2004.*, SIT 2004, pages 31–37, June 2004. doi: 10.1109/ISIT.2004.1365067.

[11] M. Gervasio, K. Myers, E. Yeh, and B. Adkins. Explanation to avert surprise. In *Explainable Smart Systems Workshop (ExSS) at ACM IUI 2018*. ACM, 2018.

[12] S. Greydanus, A. Koul, J. Dodge, and A. Fern. Visualizing and understanding Atari agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1792–1801, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[13] D. Ha and J. Schmidhuber. World models. arXiv preprint, 2018.

[14] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/hafner19a.html.

[15] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1): 53–87, Jan 2004. ISSN 1573-756X. doi: 10.1023/B:DAMI.0000005258.31418.83.

[16] B. Hayes and J. A. Shah. Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '17, pages 303–312, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4336-7. doi: 10.1145/2909824.3020233.

[17] E. Hellinger. Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *Journal für die reine und angewandte Mathematik*, 136:210–271, 1909.

[18] S. H. Huang, K. Bhatia, P. Abbeel, and A. D. Dragan. Establishing appropriate trust via critical states. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3929–3936, 2018. doi: 10.1109/IROS.2018.8593649.

[19] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912. doi: 10.1111/j.1469-8137.1912.tb05611.x.

[20] O. Z. Khan, P. Poupart, and J. P. Black. Minimal sufficient explanations for factored markov decision processes. In *Proceedings of the 19th International Conference on International Conference on Automated Planning and Scheduling*, ICAPS'09, pages 194–200. AAAI Press, 2009. ISBN 978-1-57735-406-2.

[21] A. Koul, A. Fern, and S. Greydanus. Learning finite state representations of recurrent policy networks. In *International Conference on Learning Representations*, ICLR 2019, 2019. URL https://openreview.net/forum?id=S1gOpsCctm.

[22] M. L. Littman. Memoryless policies: Theoretical limitations and practical results. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior : From Animals to Animats 3: From Animals to Animats 3*, SAB94, pages 238–245, Cambridge, MA, USA, 1994. MIT Press. ISBN 0-262-53122-4.

[23] J. McDonnell, J. Martin, D. Markant, A. Coenen, A. Rich, and T. Gureckis. psiTurk (Version 1.02) [Software]. https://github.com/NYUCCL/psiTurk, 2012. URL https://github.com/NYUCCL/psiTurk.

[24] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1 – 38, 2019. doi: 10.1016/j.artint.2018.07.007.

[25] A. W. Moore and C. G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13(1):103–130, Oct 1993. ISSN 1573-0565. doi: 10.1007/BF00993104.

[26] C. P. H. Mulder, E. Bazeley-White, P. G. Dimitrakopoulos, A. Hector, M. Scherer-Lorenzen, and B. Schmid. Species evenness and productivity in experimental plant communities. *Oikos*, 107(1):50–63, 2004. doi: 10.1111/j.0030-1299.2004.13110.x.

[27] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. ISBN 0471619779.

[28] P. Sequeira and C. Antunes. Real-Time Sensory Pattern Mining for Autonomous Agents. In *6th International Workshop on Agents and Data Mining Interaction, ADMI 2010*, ADMI 2010, pages 71–83, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15420-1. doi: 10.1007/978-3-642-15420-1_7.

[29] P. Sequeira, F. S. Melo, and A. Paiva. An Associative State-Space Metric for Learning in Factored MDPs. In *Proceedings of the 16th Portuguese Conference on Artificial Intelligence*, EPIA 2013, pages 163–174, Berlin, Heidelberg, Sep 2013. Springer Berlin Heidelberg. ISBN 978-3-642-40669-0. doi: 10.1007/978-3-642-40669-0_15.

[30] P. Sequeira, E. Yeh, and M. Gervasio. Interestingness Elements for Explainable Reinforcement Learning through Introspection. In *Joint Proceedings of the ACM IUI 2019 Workshops*, page 7. ACM, Mar 2019.

[31] R. S. Sutton and A. G. Barto. *Reinforcement learning: an introduction*. MIT Press, sep 1998. ISBN 0262193981.

[32] M. Tomczak and E. Tomczak. The need to report effect size estimates revisited. an overview of some recommended measures of effect size. *Trends in Sport Sciences*, 21(1): 19–25, 2014.

[33] J. van der Waa, J. van Diggelen, K. van den Bosch, and M. Neerincx. Contrastive Explanations for Reinforcement Learning in terms of Expected Consequences. In *Proceedings of the 2nd Workshop on Explainable Artificial Intelligence*, XAI 2018, pages 165–170, 2018.

[34] N. Wang, D. V. Pynadath, and S. G. Hill. The impact of pomdp-generated explanations on trust and performance in human-robot teams. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, AAMAS '16, pages 997–1005, Richland, SC, 2016. IFAAMAS. ISBN 978-1-4503-4239-1.

[35] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992698.

[36] T. Zahavy, N. Ben-Zrihem, and S. Mannor. Graying the black box: Understanding dqns. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1899–1908, New York, New York, USA, 20–22 Jun 2016. PMLR.

A Additional Introspection Analyses and Interestingness Elements

This appendix details the additional interestingness elements captured by our framework that were *not* included and evaluated in the user study. The complete framework operates at three levels, with the first analyzing characteristics of the task that the agent has to solve, the second behavior of the agent while interacting with the environment, and the third performs a meta-analysis combining information gathered at the lower levels. Given the number of elements generated by our framework, in the study presented in the main document subjects were exposed to only a subset of the elements. In particular, we selected elements from dimensions analyzing an agent's interaction with the environment, thereby excluding level 0 of analyses. Moreover, we focused on elements that help characterize the agents' overall behavior.

The complete framework relies on the agent collecting the following additional *interaction data*:

- $\tau(z)$ is the last (most recent) time-step of the agent's history of interaction in which z was observed. Similarly, $\tau(z, a)$ denotes the last time-step in which it executed action a after observing z;

- $\hat{R}(z, a)$ is the agent's *estimate of the reward* received for performing action a after observing z. It can be estimated by maintaining a running average of the rewards received, *i.e.*, $\hat{R}(z, a) = \sum_{t \in T_{z,a}} r / n(z, a)$, where $T_{z,a}$ contains the time-steps where action a was executed after observing z;

- $\widehat{\Delta Q}(z, a)$: the expected *prediction (Bellman) error* associated with $Q(z, a)$. Given an observed transition (z, a, r, z'), the prediction error corresponds to $\Delta Q(z, a) = r + \gamma \max_{b \in \mathcal{A}} Q(z', b) - Q(z, a)$. As with $\hat{R}(z, a)$, the agent can maintain a running average of the prediction errors by using $\widehat{\Delta Q}(z, a) = \sum_{t_{z,a}} \Delta Q(z, a) / n(z, a)$.

In addition, as is the case with many RL scenarios, we assume that, at each time-step t, the agent observes its environment through a finite set of *features* $Z_t^i = z^i, i = 1, \ldots, N$, each taking values in some feature space $\mathcal{Z}^i$. The observation space thus corresponds to the cartesian product $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_N$. When this is the case, the structure exhibited by such *factored MDPs* can also be exploited to derive interesting aspects related to specific observation elements.

Table 12 lists all the analyses implemented so far, how they are grouped at different levels, and a short description of the elements that each generates.

A.1 Level 0: Analyzing the Environment

This level of introspection analyzes characteristics of the task that the agent has to solve. The focus is therefore on the estimated reward and transition probability functions, *i.e.*, $\hat{R}$ and $\hat{P}$, respectively.

A.1.1 Transition Certainty Analysis

The estimated transition probability function $\hat{P}(z' \mid z, a)$ can be used to expose relevant aspects of the environment dynamics, in the perspective of the agent. Namely, it can be used to identify certain and uncertain transitions. Given an observation z and an action a, the transition certainty associated with (z, a) is measured according to how concentrated the observations $z' \in \mathcal{Z}$ following (z, a) are. For that purpose, we use the *evenness* measure defined in Eq. 1 to calculate the dispersion of the distribution over possible next observations according to the data stored in $\hat{P}$. Namely, for a given observation-action pair (z, a), $\xi_{za} = \xi(\hat{P}(\cdot|z, a))$ provides the evenness associated with transitioning from (z, a). The following interestingness elements are generated by this dimension of analysis:

(Un)Certain transitions: denote situations in which the next state is hard/easy to be predicted by the agent. For observations, we use ξ_{za} to identify the certainty associated with a given (z, a) pair. For features, the certainty of a feature z^i is calculated by averaging the evenness ξ_{za} for all observations $z \in \mathcal{Z}$ in which z^i is active.

Table 12: Overview of the dimensions of analysis at each level of the introspection framework and the generated interestingness elements. Dimensions and elements included for evaluation in the user study are marked in bold, and were detaile din the main document.

Level	Purpose	Dimension	Generated Interestingness Elements
0. Task	*analyze task characteristics*	Transition Certainty	(un)certain transitions: *hard/easy to predict with regards to next-step transitions*
		Reward	reward outliers: *where the agent receives a relatively high/low reward*
		Frequency	observation coverage and dispersion: *characterize the agent's exploration*; **(in)frequent situations:** *situations the agent finds very (un)common*; strongly-/weakly-associated features: *patterns in the observation features*
1. Interaction	*analyze history of interaction with environment*	**Execution Certainty**	(un)certain executions: *hard/easy to predict with regards to action execution*
		Recency	ancient situations: *experienced by the agent a long time ago*
		Value	value outliers: *where the agent expects a relatively high/low value*; mean prediction error: *how hard is the task for the agent*; prediction outliers: *situations that are "hard to learn"*
2. Meta-analysis	*combine elements from different analyses*	**Transition-Value**	**observation minima and maxima:** *learned goals and situations to avoid*
		Sequence	**most likely sequences to maxima:** *summarizes the agent's learned strategies*
		Contradiction	contradictory values and goals: *where action selection contradicts internal/external expectations*

The rationale behind the transition dimension is that transitions leading to many different states have a high dispersion and are considered *uncertain*. Likewise, transitions leading to a few states have a low dispersion can be considered *certain*. This analysis thus highlights the (un)certain elements of the agent's transitions, actions and of its observation features. Uncertain elements are especially important as people tend to resort to "abnormal" situations for the explanation of behavior [24]. This information can also be used by the agent in a more proactive manner while interacting with the environment. For example, the agent can express its confidence in the result of its actions when necessary, or request the help of a human user when it faces a very uncertain situation.

A.1.2   Reward Analysis

The idea behind this analysis is to identify uncommon situations regarding the reward received by the agent during its interaction with the environment. As $\hat{R}$ corresponds to a model of the rewards received by the agent, parts of the true reward function may have not been captured as the model will reflect the agentâĂŹs behavior in the environment. Notwithstanding, the purpose of our framework is to analyze a *particular history* of interaction rather than the fidelity of the models or the optimality of the learned behavior. The following interestingness elements are generated:

Reward outliers: correspond to observation-action pairs in which, on average among all other states and actions, the agent received significantly more or less reward. Reward outliers correspond to:

$$\forall_{(z,a)\in\mathcal{Z}\times\mathcal{A}}: |\hat{R}(z,a) - \bar{r}| > \lambda_\sigma \sigma_{\bar{r}}, \tag{3}$$

where $\bar{r}$ is the overall average reward collected by the agent, $\sigma_{\bar{r}}$ is the standard deviation of $\bar{r}$ and λ_σ is an arbitrary threshold to determine outliers. When considering observations as a whole, this information may be used to identify situations in which the agent is likely to receive relatively low or high rewards. When considering individual features, it can help capture significant individual contributions of features to the agent's reward that are otherwise "diluted" amongst all features if considering only observations.

A.2   Level 1: Analyzing the Interaction with the Environment

The purpose of this level is to help characterize the environment's dynamics, as captured by an agent, and extract important aspects of the agent's behavior and history of interaction with it.

A.2.1   Frequency Analysis

This analysis identifies interestingness elements that can be found given information stored in the frequency tables $n(z)$, $n(z,a)$ and $n(z,a,z')$. In addition to the (in)frequent situations, detailed in the main document, the following elements are extracted:

Observation coverage: corresponds to how much of the observation-space—regarding all possible combinations between the observation features—were actually observed by the agent. Formally, it corresponds to: $\sum_{z\in\mathcal{Z}} n(z)/|\mathcal{Z}|$. This is an overall performance element that can provide an indication of how much of the state-space was covered by the agentâĂŹs behavior, which is an important quality of its exploration strategy.

Observation dispersion: corresponds to how even the distribution of visits to the observation space was. In particular, it analyzes the histogram of observations using the evenness metric in Eq. 2. It can be used to infer how unbalanced the visits to states were, which in turn may denote how interesting the dynamics of the environment are, *e.g.*, denoting situations that are physically impossible to occur, or how exploratory the agent was during the interaction with it.

Strongly-/weakly-associated features: are sets of observation features (feature-sets) that frequently/rarely co-occur. To determine such elements, we resort to frequent pattern-mining (FPM) [1], a data-mining technique to find patterns of items in a set of transactions.

We first transform each observation $z \in \mathcal{Z}$ into a transaction corresponding to the set of features that are active in that observation, *i.e.*, $(z^1, z^2, \ldots, z^N)$. Each observation-transaction is then repeatedly added to a data-base for a number of times according to its frequency, as given by $n(z)$. After that, we create a frequent-pattern tree (FP-tree) [15] that facilitates the systematic discovery of frequent combinations between the items in the data-base. We use the *Jaccard index* [19], a metric that can be used to measure the association strength of item-sets [28, 29], and the FP-Growth algorithm [15] to retrieve all observation feature-sets that have a Jaccard index above or below a given threshold.[17] This element may be used to denote both patterns in the agentâĂŹs perceptions or regularities in its environment, and also rare on inexistent combinations of features. In turn, these aspects may be important to explain the agent's physical interaction with the environment and expose its perceptual limitations to an external observer.

Feature-rules: these are rules in the form $\mathsf{z}_a \Rightarrow \mathsf{z}_c$, where z_a is the rules antecedent and z_c the consequent. The idea is to determine sets of features–the antecedent—that frequently appear conditioned on the appearance of other sets of features–the consequent. We use the interest or *lift* statistical measure [1] to determine the *confidence* of every rule given the strongly-associated feature-sets. This element can be used to determine causal explanations in the environment as observed by the agent.

A.2.2   Recency Analysis

This analysis uses the data stored in the recency tables $\tau(z)$ and $\tau(z, a)$ to produce the following element:

Ancient situations: these correspond to situations that were encountered by the agent in the beginning of its interaction with the environment but that have not been visited recently. This is achieved by filtering observations and observation-action pairs whose last time-step, according to the information stored in $\tau(z)$ and $\tau(z, a)$, respectively, is below a given threshold. These elements can be useful to identify rare situations encountered by the agent, or situations that the agent tends to avoid according to its action-selection policy.

A.2.3   Value Analysis

The value functions $Q(z, a)$ and $V(z)$ learned by the agent are a very important source of interestingness. Specifically, they can be used to identify situations that are valuable for the agent, thereby denoting its goals and sub-goals when interacting with the environment. In addition, the mean prediction errors stored in $\widehat{\Delta Q}(z, a)$ can be used to identify situations in which the agent experienced learning difficulties. This analysis uses these sources of information to generate the following interestingness elements:

Value outliers: correspond to the observation- and feature-action pairs that are significantly more or less valued than other observations. We use the same outlier-detection method used outlined in Eq. 3 but using the action-value function $Q(z, a)$. This element denotes desirable situations with regards to the agentâĂŹs goals—high-value pairs denote situations conducive for the agent to attain its goals while low-valued situations might prevent the agent of fulfilling its task. If the agent's observations are composed of features, then this element can be further used to denote significant individual contributions of feature values to achieving the agentâĂŹs goals, *e.g.*, to explain which objects in the environment are more valuable to the agent or to identify which elements the agent tries to avoid.

Mean prediction error: this is the mean prediction error amongst all states and actions, corresponding to: $\sum_{z \in \mathcal{Z}} \sum_{a \in \mathcal{A}} \widehat{\Delta Q}(z, a) / |\mathcal{Z}||\mathcal{A}|$. This element is another overall performance metric, and can be used to evaluate the accuracy of the agentâĂŹs world model, *i.e.*, how well can the agent predict the consequences and future value of its actions in most of the situations it encounters. In addition, by tracking this element while the agent is learning we may verify its learning progress—if the average prediction error is decreasing over time, it means that the agent is learning the consequences of its actions. Similarly, if the value is not decreasing or is actually increasing this may mean that the agent is not learning the policy

---

[17]Further technical details can be found in [30].

properly or simply that the environment is very dynamic and unpredictable—even if only in the agentâĂŹs perspective, *i.e.*, caused by its perceptual limitations.

Prediction outliers: correspond to the observation-action pairs that have a significantly higher or lower mean prediction error associated to them. This element may denote situations that are very hard (or easy) for the agent to learn. For example, in situations where the next observation and reward received are very stochastic, the agent's future might be very unpredictable and uncertain. Exposing an end-user to such situations may help her decide that the agent is not ready for deployment and needs more training in some regions of the state space. It is also a good opportunity for the user to provide corrective feedback.

A.3   Level 2: Meta-Analysis

This level refers to analyses combining information from the different interaction data and the analyses at previous levels, resulting in the identification of more complex aspects of the interaction.

A.3.1   Transition-Value Analysis

This analysis combines information from the agent's estimated $V(z)$ function and the transition function $\hat{P}(z' \mid z, a)$. The goal is to analyze how the value attributed to some observation changes with regards to possible observations taken at the next time-step. In addition to the maxima and minima elements, detailed in the main document, the following elements are produced:

Variance outliers: Let $\bar{v}_{za} = \sum_{z' \in \mathcal{T}_{za}} |V(z) - V(z')| / |\mathcal{T}_{za}|$ be the mean absolute difference of values to the immediate observations z' taken after executing a when in z, where $\mathcal{T}_{za} \doteq \{\forall_{z' \in \mathcal{Z}} : \hat{P}(z' \mid z, a) > 0\}$ is the set of observed transitions starting from observation z and executing action a. Then, let $\sigma^2_{\bar{v}_{za}}$ denote the variance associate with the mean $\bar{v}_{za}$ for observation z and action a. For each observation $z \in \mathcal{Z}$, we then calculate $\sum_{a \in \mathcal{A}} \sigma^2_{\bar{v}_{za}} n(z, a) / n(z)$, *i.e.*, the mean difference variance among all actions, where each action is weighted according to the relative number of times it was executed. Finally, we take the standard deviation of each mean to select the observation variance outliers, *i.e.*, observations where the variance of the difference in value to possible next observations is significantly higher or lower than in other observations. This interestingness element is important in that it can be used to identify highly-unpredictable and especially risky situations, *i.e.*, in which executing actions might lead to either lower- or higher-valued next states.

A.3.2   Contradiction Analysis

This analysis combines information from the value, reward and frequency functions, the value analysis and provided domain knowledge. The goal is to identify *unexpected* situations, where the agent was expected to behave in a certain manner, but the collected data informs us otherwise. Hence, we can automatically determine the foils for behavior in specific situations.

Contradictory-values: correspond to observations in which the actions' value distribution proportionally diverges from that of their rewards. Specifically, for each observation $z \in \mathcal{Z}$ we first derive probability distributions over actions in $\mathcal{A}$ by normalizing the values and rewards associated with z, according to the data stored in $Q(z, \cdot)$ and $\hat{R}(z, \cdot)$, respectively. We then compute the Jensen-Shannon divergence (JSD) [10] that measures how (dis)similar two discrete probability distributions are with regards to the relative proportion that is attributed to each element. If the value attributed to actions is proportionally very different from the reward that the agent expects to receive by executing the same actions, the JSD will be close to 1. On contrary, low JSD values (close to 0) denote a low divergence and hence similar, or aligned, distributions.

This distinction is relevant because in typical RL scenarios, the reward is something *externally* defined by the agent's designer according to the task, while the value is something *internally* learned by the agent. Therefore, by using this technique and resorting to the information stored in Q and $\hat{R}$, we can identify situations with a *value-reward* JSD higher than a given threshold. In such situations, the agent may select actions contradicting what an

external observer would expect, *e.g.*, choosing a low-reward but high-value action. Further, we analyze the individual components of the JSD to identify which indexes are responsible for the non-alignment or dissimilarity between the distributions. This allows us to automatically detect the contradictory situations and provide justifications to users regarding the agent's unexpected behavior, *e.g.*, by showing that the selected action leads to a certain subgoal and is thus a better option compared to the expected action.

Contradictory-goals: to identify these elements, we assume that the system is provided with domain-knowledge regarding goal states. This corresponds to the agent designer's expectations over its learned behavior, *i.e.*, situations that would normally be considered as highly-desirable for the agent to perform the task by an external observer. Based on this information we determine which observations that were found by our framework to be sub-goals for the agent—identified as local maxima by the transition-value analysis—are not in the known list of goals. The goal is to avert user surprise, which can lead to the erosion of trust in the system [11]. Namely, from the contradiction list an end-user can resort to the explanation techniques to gain insights over the agent's performance, *e.g.*, visualize its behavior in contradictory situations, query about the existence of likely paths from those situations to known goals, etc.

B   Agent Experiments

This appendix provides additional details regarding the agent experiments performed in Frogger. In particular, the results of the performance in the task and of the interestingness elements are detailed for each of the three agents.

B.1   Agents Performance Results

Fig. 7 illustrates the learning performance of each agent averaged over 100 trials. As we can see, the optimized agent learns about reaching the lilypads around five eights of the training procedure and quickly learns a policy allowing it to reach high game levels and significantly decrease the number of lost lives. The high-vision agent follows a similar but less pronounced learning curve—it can survive for as long as the optimized agent but, because it takes more time crossing the road (cf. Fig. 7g), it does not reach such high levels. Contrary to these agents, the fear-water agent has a significantly poorer learning performance. As seen in Fig. 7d, it starts by surviving longer than the other two agents but when observing Figs. 7f and 7g we see that it spends most of its time on the road and in the middle grass region, eventually being hit by cars.

B.2   Introspection Results

A quantitative summary of the generated elements for each agent is listed in Table. 13. Overall, we can see that the optimized agent acquired a better understanding of the task by means of the RL training, which resulted mainly from an adequate parameterization. A first indication of that is the significantly higher number of certain sequences to sub-goals found. Additionally, this agent has more maxima and minima situations, which reflects a more solid knowledge in the situations it wants to achieve and avoid, respectively. Qualitatively, the introspection analysis for this agent determined as maxima situations where the frog is on a log facing the lilypads ($\phi_N = lilypad$) and as minima the re-start states, *i.e.*, after dying or putting a frog on a lilypad, and the bottom corners of the environment. As for execution certainty, this agent has less uncertain transitions and action executions when compared to the other agents. In particular, most of the situations occurring in the river, *i.e.*, involving jumping on logs, were considered certain while the most uncertain situations occur on the road and involve the frog being trapped by cars near the borders. As for the frequent observations, they occur both in the road—when seeing *empty* or seeing *car*—and in the river when seeing *log* in any direction.

The results for the high-vision agent are very similar to those of the optimized agent with regards to maxima, minima, and certain-execution situations. However, this agent has significantly more uncertain situations than the other agents. Namely, these occur when

Table 13: Statistics about the interestingness elements found by our framework for each agent in the Frogger task. Dimensions and elements included for evaluation in the user study are marked in bold. See the text for details on each criterion.

Dimension	Element	Criterion	Agent		
			Optim.	High-vis.	Fear-wat.
Transition Certainty	Certain trans.	$\xi_{za} \leq 0.03$	25	26	3
	Uncertain trans.	$\xi_{za} \geq 0.9$	9	20	5
Reward	Reward outliers	$\lambda_{\sigma} \geq 2.5$	9	7	2
	Obs. dispersion	—	0.78	0.78	0.73
	Obs. coverage (# unique obs.)	—	0.09 (348)	0.09 (390)	0.06 (194)
Frequency	**Frequent sit.**	$n(z) \geq 15 \times 10^3$	18	19	14
	Infrequent sit.	$n(z) \leq 150$	8	20	35
	Assoc. feat.-sets	Jacc. ≥ 0.4	5	6	4
	Feature-rules	lift ≥ 0.8	14	3	3
Execution Certainty	**Certain exec.**	$\xi_z \leq 0.1$	33	10	2
	Uncert. exec.	$\xi_z \geq 0.85$	8	13	14
Recency	Ancient sit.	$\tau(z) \leq 0.4\tau_{\max}$	2	6	16
	Value outliers	$\lambda_{\sigma} \geq 2$	8	9	2
Value	Mean pred. error	—	270 ± 277	196 ± 193	30 ± 96
	Pred. outliers	$\lambda_{\sigma} \geq 2$	12	27	6
Transition-Value	**Maxima**	—	7	2	3
	Minima	—	4	3	2
Sequence	**Certain seq.**	—	28	8	6
Contradiction	Contrad. values	—	8	7	1
	Contrad. goals	—	0	0	3

executing action N when $\phi_N = empty$ on the road near the left border—although the agent is good at seeing cars from afar, when it's near the border it cannot see cars entering the lane above, thus making it one of its most unpredictable situations.[18] As for its most frequent situations, they occur while the agent is in the road and involve observing the presence of cars, which is expected given that it spends most of its time in that region and has a very high vision range. Also, because it tries to avoid the left side of the road, this agent tends to cross it on the right side and jumps on logs mostly in the right side of the river—as a result, the less frequent situations occur when the frog is on a log near the left border.

The optimized and high-vision agents, having performed better in the task, covered more of the state space and their visits were more disperse comparing to agent fear-water. Interestingly, their mean prediction error, especially that of the optimized agent, is very high.

---

[18]The majority of car lanes in Frogger flow west to east.

This means that they were exposed to extreme situations during learning, where choosing the correct action makes a difference between receiving a very high or low reward.

The results for the fear-water agent are quite different, as expected. Because the agent was not optimistically initialized, the world from its perspective is less interesting. First, it has fewer prediction error and value outliers. Also, due to its "fear" of dying in the river, its normal behavior is to cross the road on the right until reaching the up-most car lane, at which point it tries to avoid cars by going to the left. Its learned goal (absolute maximum) occurs when observing $z = [water, car, empty, bounds]$. As for the absolute minimum it corresponds to being surrounded by cars. In addition, its uncertainty over the environment is different from that of the other agents—while it has less uncertain transitions, it has significantly more uncertain executions which results from a less certain learned policy. In particular, this agent is only certain about when to avoid cars on the road, but very uncertain when on the logs. Similarly, the frequent observations occur in the road and middle grass row. The fact that it has more infrequent and ancient situations is also related to the fact that it quickly learned to avoid jumping on logs to avoid the high penalty of falling on the river.

Finally, with regards to contradictory goals, in Frogger these correspond to all maxima observations where $\phi_N \neq lilypad$. As expected, no unexpected goals were found for both the optimized and high-vision agents, while the goals of the fear-water agent are all unexpected, thereby denoting a learned policy not contingent with the intended task.

C Survey Responses

This appendix provides in Figs. 8 and 9 the percentage of subject responses relative to each region of the Frogger environment and each agent for the region-related survey questions, corresponding to variables *time* and *practice*.

(a) Total reward per episode

(b) Game level

(c) Num. deaths

(d) Time-steps per episode

(e) Agent in the river

(f) Agent in the middle grass row

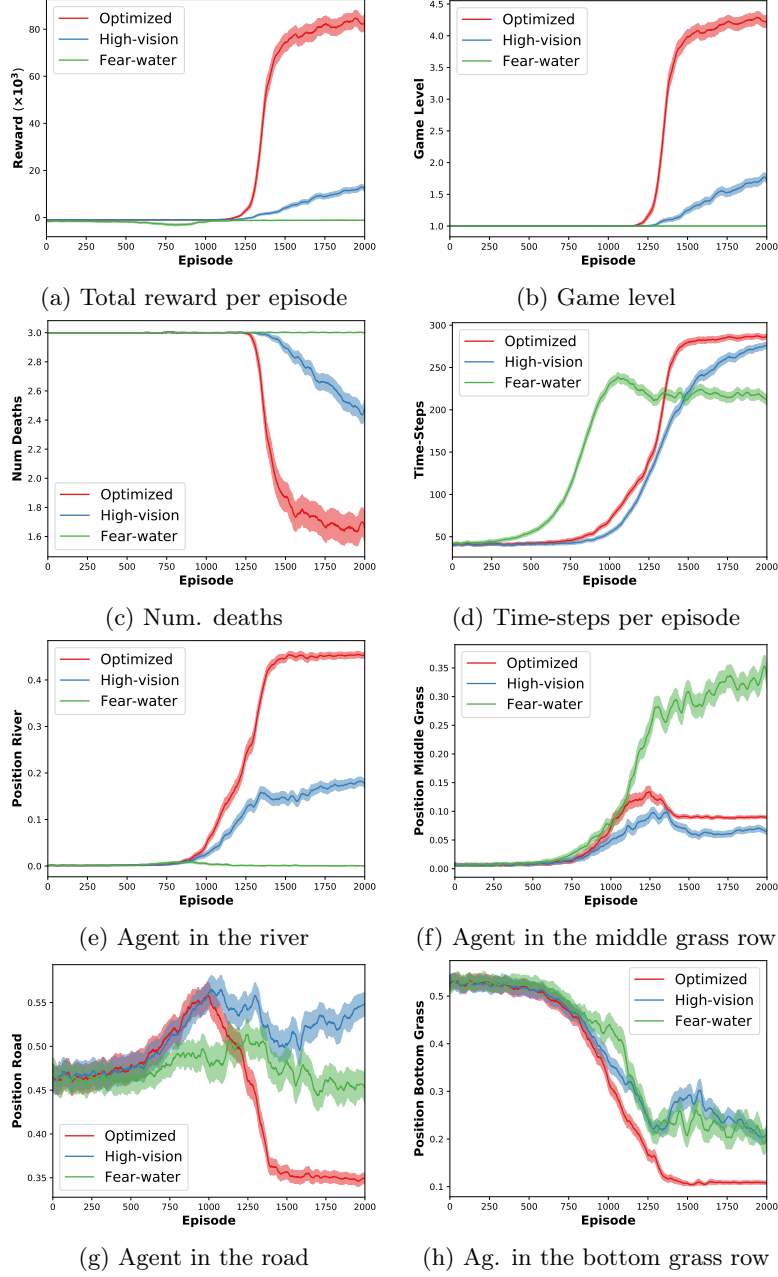(g) Agent in the road

(h) Ag. in the bottom grass row

Figure 7: Mean evolution of the agents' training procedure. Plots correspond to averages over 100 training trials of 2,000 episodes each. Shaded areas correspond to standard errors. Figs. e–h represent the mean percentage of time spent by the agent in a specific region.
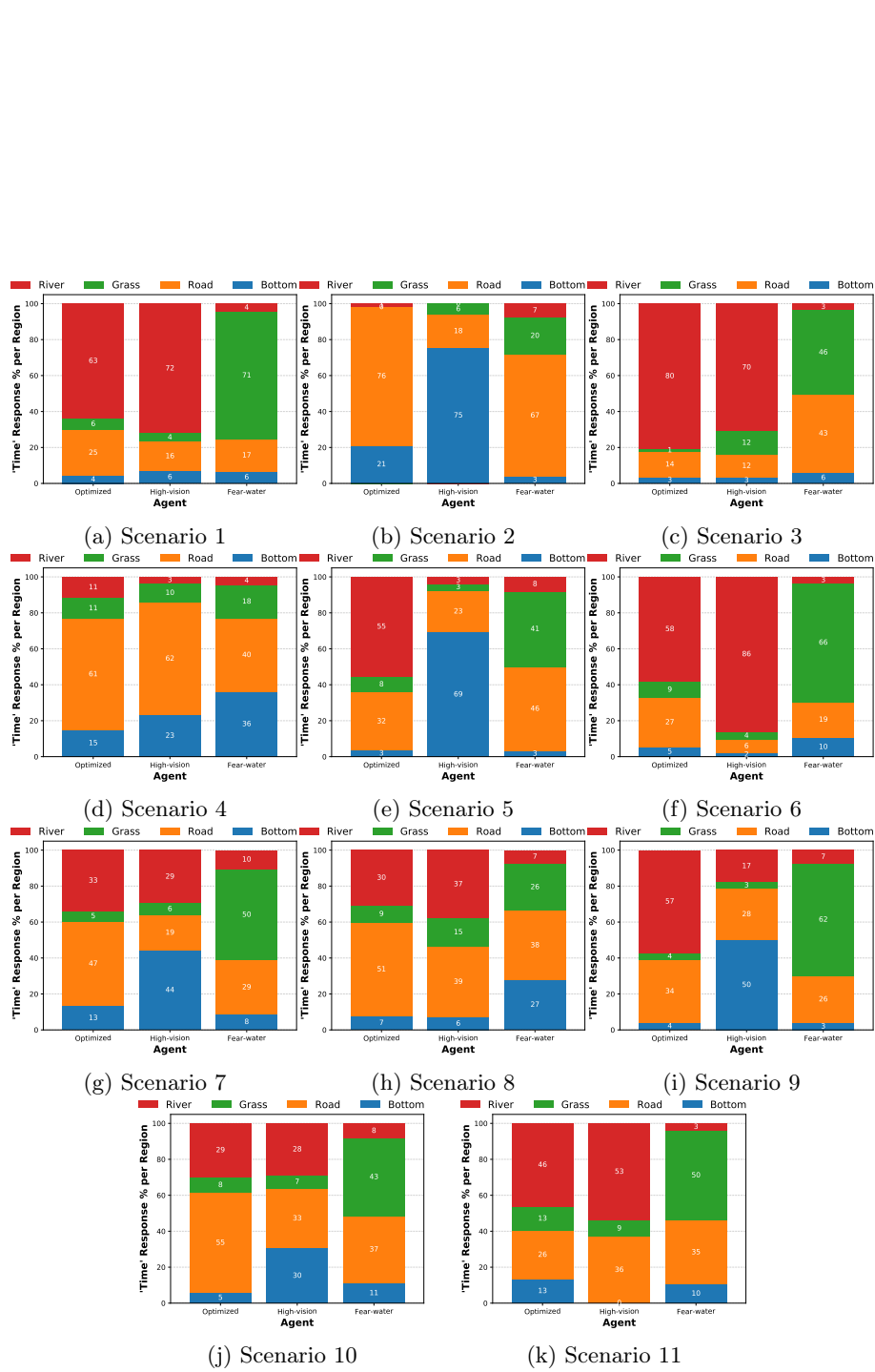
Figure 8: Subjects' responses to the *time* question, for each scenario. Plots correspond to the percentage of positive responses, *i.e.*, relative frequency that subjects selected some region for an agent in each scenario.

(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

(d) Scenario 4 (e) Scenario 5 (f) Scenario 6

(g) Scenario 7 (h) Scenario 8 (i) Scenario 9

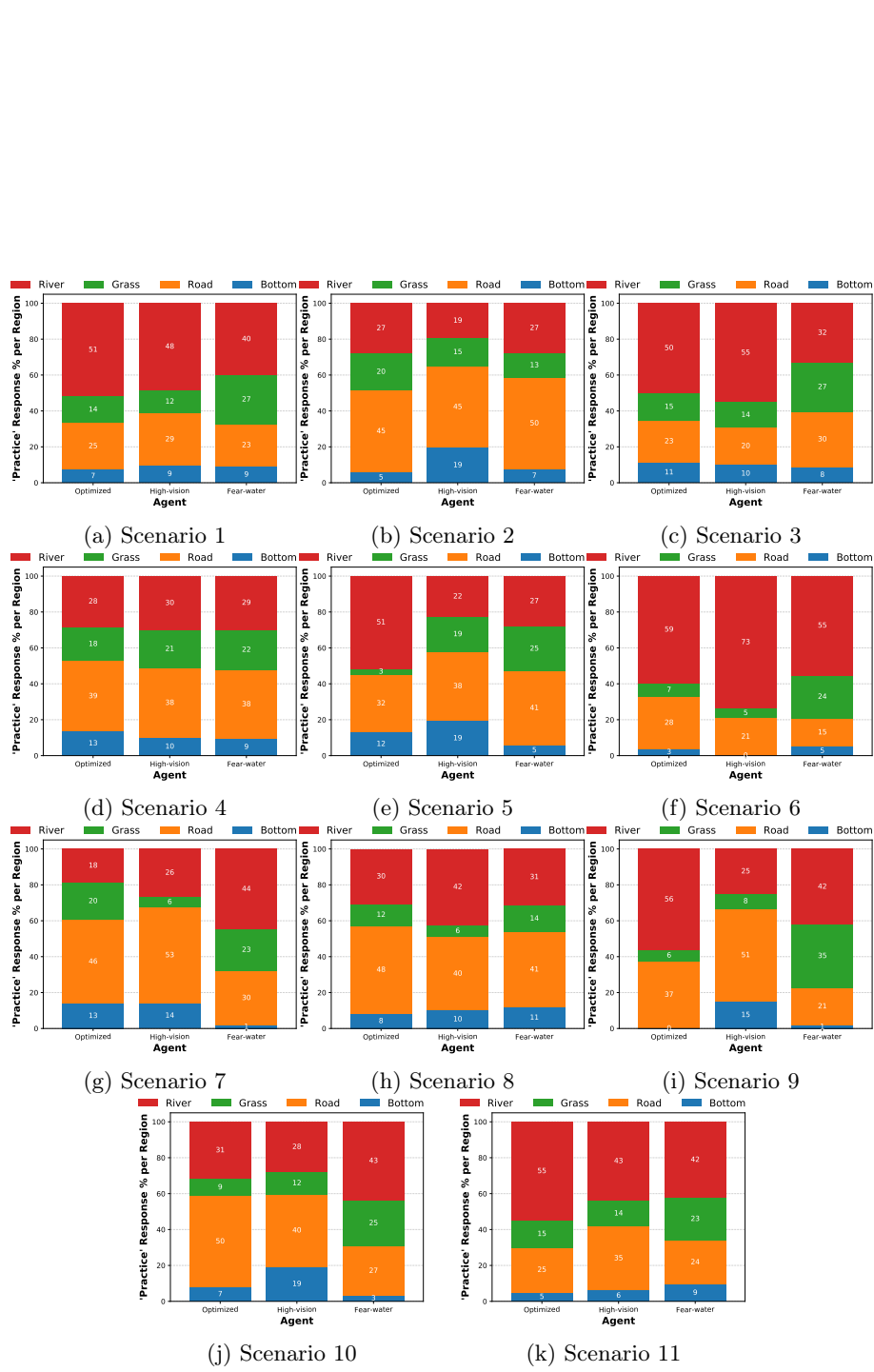(j) Scenario 10 (k) Scenario 11

Figure 9: Subjects' responses to the *practice* question, for each scenario. Plots correspond to the percentage of positive responses, *i.e.*, relative frequency that subjects selected some region for an agent in each scenario.